필터링에 기반한 고차원 색인구조의 동시성 제어 기법의 설계 및 구현

이용주° 장재우
전북대학교 컴퓨터공학과
{yjlee, jwchang}@dblab.chonbuk.ac.kr

Design and Implementation of High-dimensional Index Structure supporting Concurrency Control

Yong-Ju Lee° Jae-Woo Chang
Dept. of Computer Engineering, Chonbuk National University

요 약

최근 멀티미디어 객체를 위한 효율적인 색인 기술에 대한 많은 연구가 활발히 진행되고 있으나 이러한 색인 구조들은 단일 사용자만을 지원하는데 그치고 있는 실정이며 상용 DBMS에 통합되어 실제 응용되는 사례는 드물다. 이에 본 논문에서는 필터링에 기반한 고차원 색인구조를 위한 효율적인 동시성 제어 기법을 제안하고, 지속성 객체 시스템인 SHORE 하부 저장 구조에 통합한다. 제안하는 동시성 제어 기법은 쓰레드의 개수를 증가시켜 삽입과 검색 측면에서 실험한 결과 약30%의 응답시간 감소를 보였다.

1. 서론

최근 데이터베이스 응용에서 내용기반 검색 및 유사성에 기반한 검색에 대한 관심이 부각되고 있다. 이러한 내용기반 검색을 위해 다양한 형태의 다차원 및 고차원 색인 구조들이 제안되었다. 기존에 제시된 고차원 색인 구조에 대한 동시성 제어기법은 B-Tree, R-Tree 와 같은 트리 구조에 기반한 기법들이 대부분이다. 이는 트리 구조에서 발생할 수 있는 노드 분할, MBR 변경 연산 등에서 발생하는 동시성 저하 문제를 해결하기 위해 잠금 결합기법과 링크 기법 등이 제시되었다.

잠금 결합 기법(Lock-Coupling technique)은 트리를 순회할 때 현재 노드의 잠금을 해제하기 전에 다음에 방문할 노드에 대한 잠금을 획득한다. 그리고 노드 분할을 수행하거나 MBR 변경 연산을 상위 노드에 반영할 때는 변경에 참여하는 모든 노드들에 대한 잠금을 획득해야 한다. 이 방법은 탐색이나 삽입 연산 시에 하나의 트랜잭션이 동시에 여러 노드에 잠금을 유지해야 하므로 동시성 성능의 저하를 초래하게 된다[1].링크기법(Link technique)은 기존의 잠금 결합 기법이 커다란 동시성 저하를 초래하는 문제점을 개선하고자 각 레벨의 모든 노드들을 오른쪽 링크를 통해 연결하여 탐색 연산 시에 잠금 결합을 수행하지 않아도 상위노드에 반영되지 않은 하위 노드의 분할을 감지하여 보상할 수 있도록 한다[2].

그러나 이러한 동시성 제어 알고리즘들은 공통적으로 트리 기반의 구조이기 때문에 삽입 연산 수행 시 트리 구조 변경 연산을 수행 할 때는 동시에 트리 상의 여러 레벨에 있는 노드에 잠금을 유지하고 있어야 하므로 오버헤드가 존재하게 된다. 한편 필터링에 기반한 고차원 색인구조는 기존 트리 기반 구조의 dimensional curse 문제를 해결하여 검색 성능을 향상시킨 방법이다. 본 논문에서는 기존 단일 사용자 환경에 맞게 구현된 필터링에 기반한 고차원 색인 구조를 다수 사용자 환경으로 확장하기 위해 새로운 동시성 제어 기법을 제안한다. 제안한 동시성 제어기법은 순차적으로 파일의 마지막 레코드에 잠금을 유지하는 구조로써 트리 기반 구조에서 여러 레벨에 걸쳐 잠금을 유지하는 오버헤드가 필요치 않다.

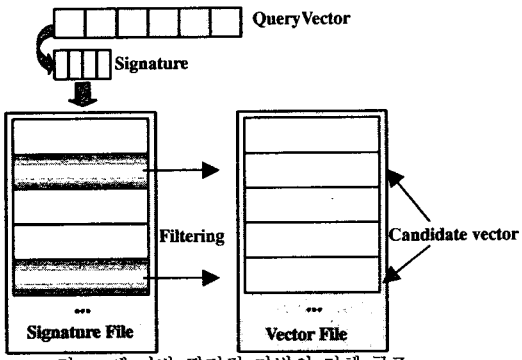
본 논문의 구성은 다음과 같다. 2 장에서는 필터링에 기반한 고차원 색인 기법을 살펴 보고, 3 장에서는 필터링에 기반한 고차원 색인 구조의 동시성 제어 기법을 제안한다. 4 장에서는 대표적인 고차원 색인 구조인 XTree 와 본 논문에서 제안한 방법을 비교 평가 하였으며, 마지막으로 5 장에서는 결론을 맺는다.

2. 필터링에 기반한 고차원 색인 기법

대부분의 고차원 색인 기법들은 차원이 증가함에 따라 급격히 성능이 저하되는 dimensional curse 문제를 가지고 있다. 이러한 문제를 해결하고자 제시된 고차원 색인 기법(CBF : Cell-Based Filtering)은 객체들의 특징벡터를 시그니처로 변환하여 특징 벡터를 직접 접근하기 전에 필터링을 수행하여 검색 성능을 향상

시킨다[3][4].

[그림 1]은 파일 시스템상에 구현된 필터링에 기반한 고차원 색인 구조의 전체적인 구조로써 질의 벡터가 주어졌을 때 이를 요약 정보를 가진 시그니쳐로 변환한다. 생성된 시그니쳐를 통해 각 셀의 범위값을 구할 수 있고 질의 벡터의 검색 범위 안에 있는 시그니쳐들 만을 다음의 검색 후보로 선택하여 필터링을 수행함으로써 검색 성능을 향상 시킨다.



[그림 1] 셀-기반 필터링 기법의 전체 구조

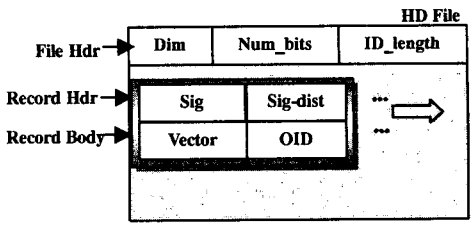
3. 필터링에 기반한 고차원 색인 기법의 동시성 제어 기법

3.1 제시하는 동시성 제어 기법

기존의 단일 사용자 환경으로 구현된 필터링에 기반한 고차원 색인구조는 물리적인 파일을 두개로 나누어 시그니쳐 정보와 벡터 정보를 저장하고 시그니쳐 정보를 메모리에 로딩하여 검색 성능을 최대한 보장하였다. 그러나 지속성 객체 저장 시스템인 SHORE(Scalable Heterogeneous Object REpository)로의 확장을 위해서는 논리적인 파일을 두개로 유지하면 오버헤드가 발생한다. 실제로 논리적 파일 두개를 사용하여 여러 개의 쓰레드가 동시에 트랜잭션을 수행할 때 서로 경합(Contention)을 일으키는 일이 발생하였다.

따라서 본 논문에서 제시한 동시성 제어기법은 SHORE 상의 하나의 논리적 파일을 두고 레코드를 순차적으로 삽입하는 구조를 채택하였으며 이를 HD(High Dimensional Index Structure)라 명명한다. 레코드의 Header 에는 기존 파일시스템의 시그니쳐 정보를 레코드의 Body 에는 벡터 파일의 정보를 유지한다. 따라서 삽입 시에는 레코드 전체에 EX-잠금을 유지하며 검색 시에는 헤더정보에 SH-잠금을 유지하는 형태를 취한다.

따라서 한 파일에 시그니쳐 정보와 벡터 정보를 모두 저장하고 HD 개방시 시그니쳐 정보를 캐시를 써서 메모리에 로딩한다. 이를 통해 일정한 만큼의 시그니쳐 정보를 탐색하여 탐색 성능을 높였으며 또한 한 파일을 사용함으로써 검색시 레코드의 Body 부분의 벡터 파일을 바로 접근할 수 있으므로 두개의 파일을 두는 방법보다 좀더 효율적인 검색이 가능하다.제안하는 HD 기법의 전체 구조는 [그림 2]와 같다.



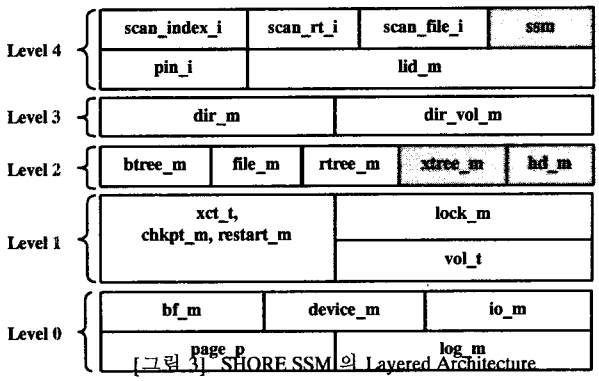
[그림 2] SHORE 상의 HD 기법의 전체구조

그러나 제안하는 방법은 실제 SHORE 상에서의 구현상 잠금(Lock)의 Escalation 발생 문제와 많은 양의 데이터를 삽입한 경우 File size 가 현저히 증가하여 성능이 저하되는 단점을 가진다.

3.2 SHORE 로의 확장

SHORE(Scalable Heterogeneous Object REpository)는 미국의 위스콘신 대학에서 개발한 지속성 객체 저장 시스템으로 기본적으로 다차원을 지원하는 rtree_m 가 존재한다. 그러나 이는 최대 4 차원의 MBR 을 저장 할 수 있는 형태이다. 이는 고차원의 특징 벡터를 저장하기에는 부적합하다. 따라서 고차원 데이터 색인을 지원하기 위해 레벨2 에 밀결합 방식으로 구현된 xtree_m 가 있다[5].

SHORE SSM 레벨2 의 기본적인 BTree, Sequential File, RTree, XTree 를 관리하는 인덱스 관리자가 존재하며 새롭게 HD 를 관리하는 인덱스 관리자를 확장하고 레벨4 에 사용자를 위한 상위레벨 API 를 설계하고 구현하였다. 확장한 SHORE 하부 저장구조는 [그림 3] 과 같다.



[그림 3] SHORE SSM의 Layered Architecture

레벨2 의 hd_m 에서의 고차원 인덱스를 실제 사용자가 사용하기 위해 구현한 레벨4 의 사용자 API 는 다음과 같다.

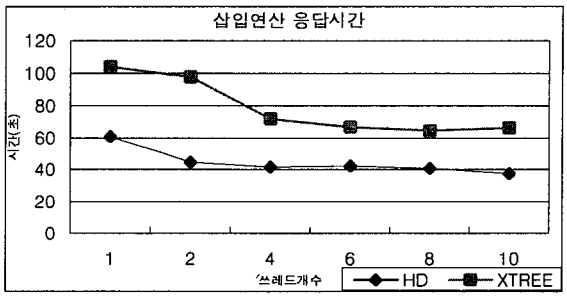
Hd_create()	- HD 인덱스를 생성한다.
Hd_destroy()	- HD 인덱스를 제거한다.
Hd_open()	- HD 인덱스를 개방한다.
Hd_close()	- HD 인덱스를 폐쇄한다.
Hd_insert()	- HD 인덱스에 벡터를 삽입한다.
Hd_delete()	- HD 인덱스의 벡터를 삭제한다.
Hd_retrieve_point()	- 포인트질의를 수행한다.
Hd_retrieve_range()	- 범위 질의를 수행한다.

Hd_retrieve_k_nearest() - K 최근접 질의를 수행한다.

4. 성능평가 및 구현 인터페이스

제안한 방법은 CPU 650 MHz dual, 메모리 128MB * 4 의 리눅스 서버에서 개발되었다. 실험을 위해 10 차원의 랜덤 데이터 2 만건을 사용하였으며 기존 SHORE 로 확장한 XTree 와 본 논문에서 제시한 HD 를 사용하여 삽입과, K-최근접 질의에 대한 성능 평가를 수행하였다.

[그림 4]는 처리량을 동일하게 조정하여 쓰레드 별 삽입 응답시간을 나타낸 것으로 XTree 는 대표적인 잠금 결합 기법을 사용한 것으며, 쓰레드 개수가 증가하면 점점 응답시간이 감소하는 것을 볼 수 있다. 본 논문에서 제시한 HD 방법은 파일의 마지막에 레코드를 삽입하는 구조로 쓰레드별 응답시간이 감소하는 것을 볼 수 있으며 성능은 쓰레드 개수가 4 일때 42% 정도의 응답시간이 감소된다. 아울러 전체적으로는 XTree 보다 약 30%의 응답시간 감소를 보인다.

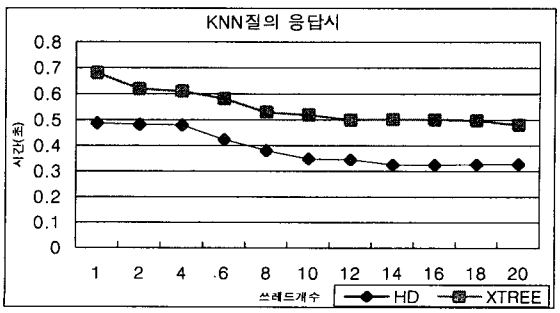


[그림 4] 삽입연산에서의 응답시간

[그림 5]는 처리량을 동일하게 조정하여 K-최근접 질의를 수행 한 것으로 K 값은 10 으로 고정하고 쓰레드별 개수가 2,..,20 개로 증가시키면서 실험한 결과를 나타낸다. 쓰레드 개수가 증가할수록 점점 응답시간이 줄어드는 것을 볼 수 있으며 쓰레드 개수가 4 일때 21%의 응답시간이 감소된다. 아울러 전체적으로 XTree 보다 약30%의 응답시간 감소를 보인다.

[그림 5] K-최근접 질의의 응답시간

[그림 6]의 질의 인터페이스는 SHORE 하부 저장 구조를 이용하여 자바 애플릿으로 만든 예제이다. 6000 개의 자연 풍경 이미지에서 15 차원의 칼라 특징 벡터를 추출하여 임의의 이미지 30 개를 보여 주고 그 이미지와 유사한 색상의 이미지를 찾는



그림이다. [그림 6]의 오른쪽 그림은 선택한 이미지를 이용하여 K-최근접 질의를 수행한 결과이다. 동시에 두개의 쓰레드가 동작하여 작업을 수행한다.

[그림 6] 유사성 질의 시스템 인터페이스

5. 결론

필터링에 기반한 고차원 색인구조는 기존의 트리에 기반한 고차원 색인구조의 dimensional curse 문제를 해결하고자 벡터 공간을 일정한 크기의 셀로 분할하여 시그니쳐 정보를 유지하고 이를 이용하여 탐색을 함으로써 검색의 성능을 향상시키는 방법이다. 본 논문에서는 기존의 단일 사용자 환경에 맞게 구현된 필터링에 기반한 고차원 색인 구조를 다수 사용자 환경으로 확장하기 위해 대표적인 지속성 객체 저장 시스템인 SHORE 로 확장하고 기존의 대표적 고차원 색인 구조인 XTree 와 성능을 비교하였다.

향후 연구 과제로는 회복 기법의 설계 및 구현을 하는 것이다.

참고문헌

[1] M Kornacker and D.Banks, "High-Concurrency Lockings in R-Trees", In Proc. 21st Int'l Conference on VLDB, pages 134-145, September 1995.

[2] M. Kornacker, C. Mohan and J.M. Hellerstein, "Concurrency and Recovery in Generalized Search Trees", In Proc ACM SIGMOD Conf., Pages 62-72, May 1997.

[3] Roger Weber, Stephen Blott," An Approximation-Based Data Structure for Similarity Search", Technical report Nr. 24, ESPRIT project HERMES(no.8141), October 1997.

[4] 한성근, 이용주, 장재우, 김현진, "효율적인 고차원 데이터 색인을 위한 셀-기반 필터링 기법", Proceedings of Korean Database Conference 2000, pp.26-35, 2000. 5.

[5] 정재욱, 장재우, "멀티미디어 응용을 위한 SHORE 하부 저장 시스템의 확장", 한국정보과학회 '99 가을 학술발표 논문집(1), pp. 6-8, 1999,10.