분산 데이터베이스 시스템에서의 색인 구성비용 절감을 위한 효율적인 색인 전송기법

°박상근* 김호석* 이재동** 배해영*
*인하대학교 전자계산공학과, **단국대학교 전산과
°skpark@dblab.inha.ac.kr

An Efficient Index Transfer Method for Reducing Index Organization Cost In Distributed Database Systems

Sang-Keun Park° Ho-Seok Kim Hae-Young Bae
School of Computer Science & Engineering, Inha University

요 약

분산 데이터베이스 시스템 환경에서는 특정 노드로 집중되는 부하의 분산이나 가용성 및 안정성 제공을 위해 데이터 분할기법 (fragmentation)과 복제기법(replication)을 사용한다. 이때 전송된 데이터에 대한 기존의 색인 재활용 기법과 벌크 로딩(bulk loading) 기법은 효율적인 색인 구성을 위해 논리적인 페이지 포인터를 물리적 주소로 변환하는 물리적 사상구조를 필요로 하거나, 색인 구성시간과 검색성능 모두를 향상시키지 못하는 문제점을 지닌다. 본 논문에서는 이와 같은 문제점을 해결하기 위해 색인 전송기법을 제안한다. 본 기법은 색인 재활용을 위해 물리적 사상구조를 추가로 유지하거나, 검색 성능을 향상시키기 위해 전체 데이터 집합을 정렬하는 것이 아니라, 데이터가 전송될 사이트에 색인구조를 저장하기 위한 물리적 공간을 예약하고 예약된 공간에 색인구조를 전송, 기록함으로써 색인 구성비용을 줄이게 된다. 또한 예약된 공간을 연속적인 페이지구조로 구성함으로써 색인 구성 시 자식노드에 대한 위치정보를 예상하여 부모노드가 지니는 자식노드에 대한 위치정보 기록 비용을 줄일 수 있다.

1. 서 론

분산 데이터베이스 시스템 환경에서는 데이터 특성에 따른 관리의 목적이나 특정 노드로 집중되는 부하의 분산 외에 가용성이나 물리적 데이터의 안정성 보장을 위해 데이터를 분할(fragmentation)하거나 복제(replication)한다. 이 때 전송된 대량의 데이터에 대한 색인 구성시간이나 검색 성능 개선을 위해 색인 재활용 기법이나 벌크 로딩 기법을 사용한다. 기존의 색인 재활용 기법은 페이지 오프셋(page offset)과 슬롯 오프셋(slot offset)으로 이루어진 튜플 ID의 물리적 사상을 위해 논리적 페이지 번호와 물리적 페이지 번호의 쌍으로 구성된 변환 테이블(translation table)을 이용한다[1,2]. 다차원 색인의 벌크 로딩 기법은 특정 차원에 대한 정렬 알고리즘이나 힐버트 값(Hilbert value)을 이용한 힐버트 정렬(Hilbert Sorting) 알고리즘 등을 이용하여 유사도 검색에 대한 성능만을 향상시키거나, 트리의 중간노드에 버퍼를 두어 동시에 다중 삽입이 가능하게 함으로써 색인 구성 시간만을 단축시켰다[3,10].

기존의 알고리즘들 중 색인 재활용 기법은 물리적 사상을 제공하는 변환 테이블을 유지하기 위한 추가비용이 요구되며, Nearest-X 알고리즘[4]과 힐버트 정렬 알고리즘 등은 검색 성능은 향상되나 데이터의 특성에 따른 정렬비용으로 인해 색인 구성 시간이 증가된다. 버퍼 트리[5]를 이용한 벌크 로딩은 색인 구성 시간은 단축할 수 있으나 데이터의 특성을 이용하지 못했기 때문에 검색 성능이 저하된다[6]. 또한 데이터 삽입을 위한 검색 비용, 노드의 오버플로우에 의한 분할 비용, 특히 공간 데이터의 경우 MBR의 조정(adjustment)[8] 등의 비용은 여전히 존재하게 된다.

본 논문[1]에서는 이와 같은 문제점을 해결하기 위해 효율적인 색인 전송기법을 제안한다. 본 기법은 전송된 데이터의 색인을 구성하기 위해 물리적 사상구조를 유지하거나 데이터의 특성에 따라 정렬하는 것이 아니라 데이터가 전송될 사이트에 색인 구성을 위한 저장공간을 예약한 후, 노드를 페이지 단위로 전송, 기록함으로써 색인 구성 시 발생하는 검색, 분할 및 MBR 조정 비용을 줄이게 되며, 잎노드의 경우 노드 내 아이템과 아이템이 가리키는 튜플을 함께 전송하여 튜플 기록 후 얻어진 튜플 ID를 아이템에 기록함으로써 정렬비용과 물리적 사상문제를 해결하였다. 또한 색인구조 저장을 위한 예약공간을 순차적인 페이지구조로 구성함으로써 색인 구성 시 기록되는 자식노드에 대한 위치 정보를 예상하여, 자식노드에 대한 물리적 위치정보를 노드 페이지 기록과 동시에 기록함으로써 자식노드 위치정보 기록을 위한 부모노드 접근 비용을 제거하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 색인 재활용 기법과 벌크 로딩 기법에 대해 기술하고, 3장에서는 본 기법 적용을 위한 분산 공간 데이터베이스 관리 시스템의 데이터베이스 구조에 대해 설명한다. 4장에서는 본 논문에서 제안된 노드간 위치정보 기록 알고리즘 및 색인 전송 알고리즘에 대해 설명하며, 5장에서는 분산 공간 데이터베이스 시스템 환경에서 기존의 벌크 로딩 기법과 본 논문에서 제안한 색인 전송 기법에 대한 성능평가를 실시한 후, 마지막으로 6장에서 결론을 맺기로 한다.

2. 관련연구

2.1 기존의 색인 재활용 기법

분산 데이터베이스 관리 시스템인 Mariposa[11]는 전송된 색인의 논리적 페이지 포인터와 물리적 주소의 사상을 위해 논리적 페이지 번호와 물리적 페이지 번호의 쌍으로 구성된 변환 테이블을 사용하였다. 변환 테이블은 포인터 값 변환을 한번의 접근으로 해결하며, swizzle 포인터 값 기록을 위해 모든 페이지의 포인터 값의 갱신을 피했다[11]. 그러나 페이지 포인터 변환기법은 물리적 사상을 위해 변환 테이블의 생성, 검색 등의 유지비용을 필요로 하며 레벨이 추가되는 경우 변환 테이블 내 기록에 대한 문제점이 존재하게 된다.

2.2 기존의 다차원 색인구조 벌크 로딩 기법

다량의 다차원 데이터에 대한 효율적 색인 구성 기법인 Hilbert R-tree construction method는 space-filling curves를 이용, 점을 Hilbert 값에 따라 정렬함으로써 공간에 대한 이웃관계를 유지하여 검색 성능을 높인다[3]. 그러나 공간객체는 Hilbert 값이 아닌 MBR에 의해 표현되기 때문에 차원이 높은 객체에 대한 겹침으로 인해 검색 성능을 저하시키는 문제점이 존재한다.

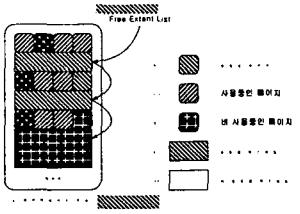
다양한 삽입 알고리즘에 대하여 구성 성능을 증가시키기 위한 기술인 버퍼 트리를 이용한 벌크 로딩 알고리즘은 트리의 중간노드에 버퍼를 두

[1] 본 연구는 대학 IT연구센터 육성.지원사업의 연구 결과로 수행되었음

어 다중 삽입이 가능하게 하였으며 overflow 발생 시 삽입 프로세스가 트리의 하위 레벨로 전이되어 색인 구성시간을 단축시킨다. 그러나 데이터의 특성을 고려한 declustering을 하지 않아 검색 성능이 저하되는 문제점을 지닌다.

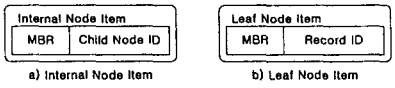
3. 데이터베이스 및 색인구조

본 기법을 적용한 분산공간 데이터베이스 관리시스템의 환경을 위한 최소 구성요소는 메타정보 관리를 위한 네임서버(name server), 분산 환경 내 데이터 상호 조정을 위한 Coordinator, 지역 서비스를 유지하며 분산 환경 내 하나의 노드로 제공되는 지역 사이트(local site), 지역 사이트의 지역 자치성 보장과 분산 서비스 제공을 위한 지역 사이트 관리기(local site manager)로 구성된다. 본 환경의 데이터베이스 구조는 [그림3-1]과 같다.



[그림3-1] 데이터베이스 구조

본 환경에서는 색인의 노드로 파일 I/O의 기본 단위인 페이지를 사용하며, 4개의 페이지는 데이터베이스 공간할당의 기본 단위인 익스텐트(extent)를 구성한다[1]. Free Extent List는 데이터베이스 내 테이블이나, 색인구조 등에 사용되었던 익스텐트 중 반환된 익스텐트들의 리스트를 나타낸다. b)는 기록되었던 데이터가 무효하게 되어 반환된 페이지, c)는 사용중인 페이지, d)는 비 사용중인 페이지, e)는 기록되었던 데이터가 무효하게 되어 반환된 익스텐트, f)는 미 할당된 익스텐트로 현재 반환되었거나 사용중인 익스텐트 중 마지막 익스텐트의 다음 영역을 의미한다. 본 기법은 전송된 색인구조 저장을 위해 연속적인 f)를 예약, 사용함으로써 색인 내 노드들에 대한 오프셋(offset) 값의 연속성을 보장한다.



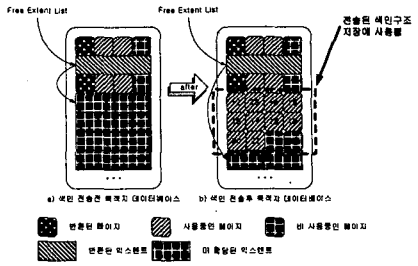
[그림3-2] R-Tree 중간노드와 잎노드 아이템

본 환경에서의 색인은 익스텐트의 리스트 구조를 이용, 데이터를 저장하며 [그림3-2] a)는 본 기법을 적용한 R-Tree의 중간노드에 저장되는 아이템(item)을 나타낸다. [그림3-2] a)는 자식노드 내 객체들에 대한 최소경계사각형과 자식노드 ID 값을 지니며, b)는 잎노드에 저장되는 아이템으로 실제 저장된 공간객체에 대한 최소경계사각형과 공간객체 튜플 ID 값을 지닌다.

4. 색인 전송 기법

4.1 자식노드 위치정보 기록 알고리즘

색인구조를 전송하여 기록하는 경우, 전송 이전의 중간노드 내 기록된 자식노드 ID는 물리적 저장구조의 특성에 의해 전송 후 기록될 데이터베이스에서 사용할 수 없다. 본 기법은 이러한 물리적 저장구조의 사상문제를 해결하기 위해 [그림4-1] a)의 미 할당된 연속적인 익스텐트를 필요한 색인노드 개수만큼 예약함으로써 노드들의 ID인 페이지 오프셋 값을 예측, 중간노드 내 아이템들이 지니고 있는 자식노드 ID 값을 결정하여 자식노드 ID 기록을 위한 부모노드로의 연속적인 차후 접근 비용을 제거하였다.



[그림4-1] 색인구조가 10개의 노드로 구성된 경우

[그림4-1]은 색인구조가 전송되기 전과 후의 목적지 데이터베이스 구조를 나타내며, 전송 중 자식노드 ID 기록과정은 다음과 같다.

[가정]

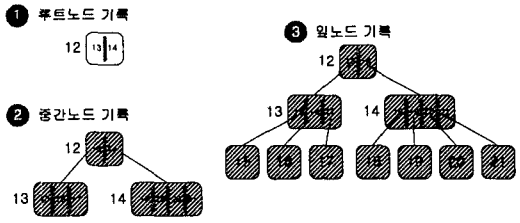
위치: 색인이 저장될 사이트, [그림4-1] a)

- 현재 사용 중이거나 반환된 익스텐트 ID는 0, 1, 2
- Free Extent List 는 익스텐트 ID가 1→3 으로 구성됨
- 데이터와 색인구조를 전송할 타 사이트가 지니고 있는 색인 구조는 10개의 노드로 이루어짐

[기록과정]

위치: 색인이 저장될 사이트

- 타 사이트로부터 10개의 노드로 구성된 색인이 전송된다는 메시지를 받음
- 10개의 페이지 저장공간을 위해, 미 할당된 순차적 익스텐트인, 익스텐트 ID 3, 4, 5번을 예약
- 익스텐트 ID 3, 4, 5번이 지니고 있는 페이지 오프셋 12~21은, [그림4-2]와 같은 구조로 자식노드의 페이지 ID로 기록



[그림4-2] 자식노드 페이지 ID 기록

[그림4-2] ①의 12번 노드는 2개의 아이템을 가지고 있으며 각 아이템이 지닐 자식노드 ID인, 페이지 오프셋 값은 [그림4~1]에 의해 13, 14임을 예상할 수 있다.

본 기법은 이와 같은 과정을 거쳐 자식노드 ID를 예상함으로써, 색인구조 저장 시 기록되어야 할 자식노드 ID를 자식노드 할당 이전에 기록함으로써 자식노드 기록을 위한 부모노드의 접근비용을 제거하였다.

4.2 색인 전송 알고리즘

전송할 데이터와 색인을 지니고 있는 지역 사이트가 A이고 데이터를 전송받을 지역 사이트가 B이면 전송과정은 다음과 같다.

[가정]

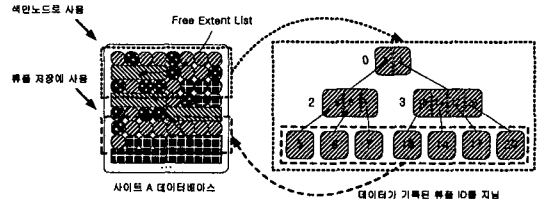
- 전송전 사이트 A에 저장된 튜플과 색인구조는 [그림4-3] a)와 같이 노드 ID의 연속성을 보장하지 않는 여러 개의 익스텐트 구조를 사용해 저장

[전송과정]

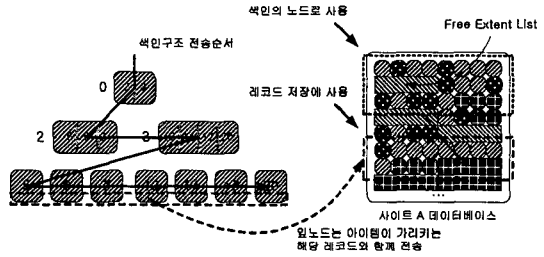
- 관리자가 사이트 A에 저장된 특정 테이블을 사이트 B로 옮기라는 메시지를 사이트 A에 전송
- 사이트 A는 색인에 사용된 노드 개수를 포함한 색인구조 저장공간 예약 메시지를 B에 전송
- 사이트 B에서 예약완료 응답 메시지 전송
- 사이트 A는 [그림4-3] b)와 같은 순서로, 루트노드부터 중간노드, 잎노드 순으

로 색인구조를 전송

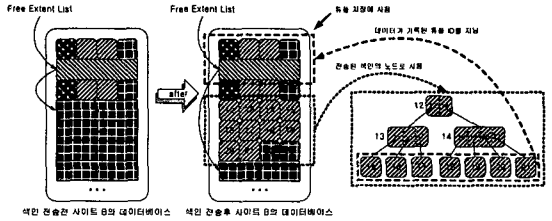
- 잎노드 전송시 사이트 A 색인구조의 잎노드에 기록된 튜플 ID 값은 물리적 특성에 의해 사이트 B에서는 사용 불가, 잎노드 전송시 잎노드 내 아이템이 가리키는 해당 튜플과 잎노드 아이템을 함께 전송
- 색인구조를 전송받은 사이트 B는 [그림4-3] c)와 같이 색인 구조를 기록하며, 전송받은 잎노드 내 아이템이 지니는 튜플 ID는, 전송된 해당 튜플을 먼저 기록한 후, 이때 발생하는 튜플 ID로 교체 기록한다



a) 사이트 A에 구축된 색인과 테이블



b) 사이트 A의 색인구조 전송순서



c) 사이트 B로 전송되어 저장된 색인과 테이블

[그림4-3] 색인 전송과정

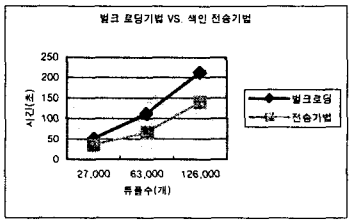
5. 성능평가

본 장에서는 분산 공간 데이터베이스 관리 시스템 환경에서 기존의 벌크로딩 기법[4]과 본 논문에서 제안하는 색인 전송 기법에 대한 성능평가를 하였다. 테스트 환경은 [표5-1]과 같다.

시스템 속성			
네트워크 속성	100M bps		
네임서버(1대)	Pentium IV 1.8 CPU 512M memory	분산 환경 내 메타데이터 관리	
상호 조정자(1대)		데이터 상호조정	
지역 사이트(2대)		분산 및 지역 서비스 제공	
데이터 속성			
데이터 종류	색인	객체명	색인 구성노드 개수
수원시 지도 데이터	R-tree	상호명	108개
		기초번호	212개
		건물	423개

[표5-1] 실험 환경

환경 내 데이터 전송과 색인 구성을 위한 주요 요소는 전송을 위한 data scan, 전송 시간, 데이터 기록 시간, 색인 구성 시간을 들 수 있다. 테스트 결과 [표5-2]과 같이 색인 전송기법이 기존의 색인 생성기법에 비해 약 30% 정도 개선된 성능을 보이며 데이터의 양이 증가할수록 편차가 커짐을 확인할 수 있다.



[표5-2] Nearest-X 알고리즘 VS. 색인 전송기법

6. 결론 및 향후 연구과제

본 논문에서 제시한 색인 전송 기법은 전송될 데이터의 색인 구조를 페이지 단위로 전송, 기록함으로써 색인 구축을 위한 정렬비용, 검색비용, 분할비용, MBR 영역의 조정 비용을 절감한 색인 구성 기법이다. 또한 부모노드 내 기록될 자식노드에 대한 위치정보를 연속적인 물리공간으로 예약하여 자식노드 할당 이전에 ID를 예상, 기록함으로써 자식노드 위치정보 기록을 위한 부모노드로의 차후 접근비용을 제거하였다. 그 밖에 잎노드 전송 시 튜플 ID 기록을 위해 잎노드와 튜플을 같이 전송하여 선 튜플 삽입, 후 잎노드 아이템 기록으로 튜플 ID에 대한 사상문제를 해결하였다.

향후 연구 과제로는 색인의 범위(range) 전송과 색인 전송 중 발생하는 실패에 대한 회복기법 방안이 연구 중에 있다.

참고문헌

[1] 박상근, 박순영, 정원일, 김영근, 배해영, " GMS: 공간 데이터베이스 관리 시스템" , 개방형GIS학회, 2003. 03

[2] Paul M. Aoki, " Recycling Secondary Index Structures" , Sequoia 2000 Technical Report No. 95-66, July 1995

[3] Ibrahim Kamel, Christos Faloutsos, " On Packing R-tree" , CIKM, pp.490-499, 1993

[4] N. Roussopoulos, D. Leifker, "Direct spatial search on pictorial databases using packed r-trees", Proc. ACM SIGMOD, May 1985

[5] Arge L, " The Buffer Tree: A New Technique for Optimal I/O-Algorithms" WADS, pp.334-345, 1995

[6] 복경수, 이석희, 조기형, 유재수, " 고차원 색인 구조를 위한 효율적인 벌크 로딩" , 한국정보처리학회, 2000.08

[7] Lars Arge, Klaus H. Hinrichs, Jan Vahrenhold, Jeffrey S. Vitter, " Efficient Bulk Operations on Dynamic R-trees" , In Proceedings ALENEX '99, 1999

[8] A. Guttman, " R-Trees: A Dynamic Index Structure for Spatial Searching" , Proc. ACM SIGMOD , pp. 47-57 , June 1984

[9] Christian Böhm, Hans-Peter Kriegel, " Efficient Bulk Loading of Large High-Dimensional Indexes" , Proc. Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'99), Florence, Italy, 1999

[10] Van den Bercken J, Seeger B, Widmayer, " A General Approach to Bulk Loading Multidimensional Index Structures" , VLDB Conference, pp.406-415, 1997

[11] M. Stonebraker, P. M. Aoki, R. Devine, W. Litwin and M. Olson, " Mariposa: A New Architecture for Distributed Data," Sequoia 2000 Tech. Rep. 93/31, Univ. of California, Berkeley, CA, May 1993.