공간 데이터 웨어하우스에서 부분 색인 전송을 이용한 효율적인 색인 재구성 기법

°정영철*, 유병섭*, 박순영*, 이재동**, 배해영*
*인하대학교 컴퓨터・정보 공학과
**단국대학교 정보컴퓨터학부
e-mail : {°aslongas3, subi, sunny}@dblab.inha.ac.kr, letsdoit@dankook.ac.kr, hybae@inha.ac.kr

An Efficient Method of the Index Reorganization using Partial Index Transfer in Spatial Data Warehouses

°Young-Cheol Jeong*, Byeong-Seob You*, Soon-Young Park*, Jae-Dong Lee**, Hae-Young Bae*
*Dept. of Computer Science & Information Engineering, Inha University
**Division of Information and Computer Science, Dankook University

요 약

공간 데이터 웨어하우스 구축기는 소스 데이터의 변경 사항을 일괄처리의 형태로 공간 데이터 웨어하우스에 적재한다. 또한, 공간 데이터 웨어하우스 서버는 사용자의 질의에 빠른 응답을 하기위해 적재된 데이터로 색인을 구축한다. 색인을 구성하는 기존 기법으로는 벌크 삽입 기법 및 색인 전송 기법이 있다. 벌크 삽입 기법은 색인을 구성하기 위한 클러스터링 비용이 필요하며 검색 성능도 떨어진다. 또한, 색인 전송 기법은 주기적인 소스 데이터의 변경을 지원하지 않는다는 문제점이 있다.

본 논문[1]에서는 이와 같은 문제점을 해결하기 위해 공간 데이터 웨어하우스에서 부분 색인 전송을 이용한 효율적인 색인 재구성 기법을 제안한다. 제안 기법은 구축기에서 색인의 구조에 맞게 클러스터링된 클러스터들을 부분 색인으로 구성하여 페이지 단위로 전송한다. 공간 데이터 웨어하우스 서버에서는 전송된 부분 색인의 물리적 사상 문제를 해결하기 위해 물리적으로 연속된 공간을 예약하고 예약된 공간에 부분 색인을 기록한다. 기록된 부분 색인은 공간 데이터 웨어하우스 서버에 있던 기존 색인에 삽입된다. 부분 색인이 기존 색인에 직접 삽입됨으로써 색인 재구성을 위한 검색, 분할, 재조정 비용은 최소가 된다.

1. 서론

공간 데이터 웨어하우스는 지리정보를 주제 중심적이고 통합적이며 시간성을 가지는 비 휘발성 자료로 저장하여 효율적인 의사결정을 지원하는 시스템이다[1]. 공간 데이터 웨어하우스는 공간 데이터 웨어하우스 서버와 구축기로 구성된다. 구축기는 소스 데이터의 변경 사항을 구축기 내부의 저장소에 적재하며, 적재된 데이터를 일정 주기마다 일괄처리의 형태로 공간 데이터 웨어하우스에 적재한다[2]. 또한, 공간 데이터 웨어하우스 서버는 사용자 서비스를 정지하고, 구축기로부터 전송되어 지는 데이터를 적재하며 사용자의 질의에 빠른 응답을 하기위해 적재 된 데이터로 색인을 구축한다. 색인을 구성하는 기존 기법으로는 벌크 삽입 기법 및 색인 전송 기법이 있다. 벌크 삽입 기법은 색인을 구성하기 위한 클러스터링 비용이 필요하며 검색 성능도 떨어진다. 또한, 색인 전송 기법은 주기적인 소스 데이터의 변경을 지원하지 않는다는 문제점이 있다[3, 4].

본 논문에서는 이와 같은 문제점을 해결하기 위해 부분 색인 전송을 이용한 효율적인 색인 재구성 기법을 제안한다. 본 기법은 구축기에서 추출된 데이터를 공간의 근접도가 아닌 색인의 구조에 맞게 클러스터링하며, 생성된 각 클러스터로 부분 색인을 생성하여 페이지 단위로 전송한다. 공간 데이터 웨어하우스 서버에서는 전송된 부분 색인의 물리적 사상 문제를 해결하기 위해 물리적으로 연속된 공간을 예약하고 예약된 공간에 부분 색인을 기록한다. 또한, 기록된 부분 색인은 공간 데이터 웨어하우스 서버에 있던 기존 색인에 일반적인 삽입 알고리즘[5]으로 삽입된다. 따라서 본 기법은 부분 색인이 기존 색인에 직접 삽입됨으로써 색인 재구성을 위한 검색, 분할, 재조정 비용을 최소로 줄일 수 있다. 또한, 공간 또는 비공간 데이터의 색인 재구성 비용 절감을 위한 다수의 트리 계열 색인 구조에 적용할 수 있으며 특히, 공간 객체 삽입 시 부모 노드 엔트리의 재구성 비용이 높은 R-Tree[5] 계열 색인의 재구성 비용 절감에 보다 효율적이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로

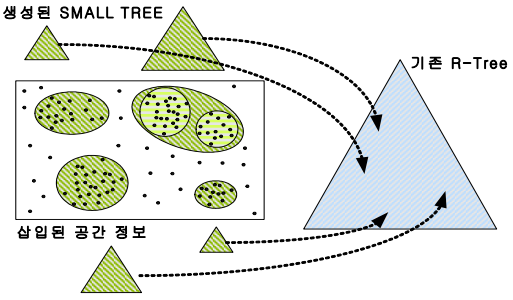
1) 본 연구는 대학 IT연구센터 육성•지원사업의 연구결과로 수행되었음

다차원 색인의 벌크 삽입 기법, Seeded 클러스터링 기법 및 색인 전송을 통한 색인 구성 기법에 대해 설명하고, 3장에서는 본 논문에서 제안하는 부분 색인 전송을 이용한 색인 재구성 기법을 설명한다. 4장에서는 부분 색인 전송 기법을 이용한 색인 재구성 기법의 성능평가를 한다. 마지막으로, 5장에서는 결론을 맺는다.

2. 관련연구

2.1 다차원 색인의 벌크 삽입

다차원 색인의 벌크 삽입 기법은 (그림 1)과 같이 삽입 데이터를 공간상에서 근접한 데이터끼리 클러스터링하여 여러 개의 클러스터를 생성한다[4, 6]. 생성된 각 클러스터는 Small Tree로 구성되며 각 Small Tree는 기존 R-Tree에 삽입된다[7]. 또한, 어떤 클러스터에도 포함되지 않는 데이터들은 열외자(outlier)로 분류하여 일반적인 삽입 알고리즘[5]에 의해 하나씩 삽입한다[4]. 이 기법은 데이터를 클러스터링 하는 비용이 많이 들며, 기존 R-Tree 노드들과 새로 삽입되는 Small-Tree 노드들 간의 겹침이 크다. 따라서 벌크 삽입 기법은 삽입 성능 및 검색 성능이 떨어지는 단점이 있다.



(그림 1) 벌크 삽입

2.2 Seeded 클러스터링 기법

Seeded 클러스터링 기법은 기존의 공간 근접도에 따른 클러스터링 방법이 아닌 이미 구축되어 있는 색인의 구조에 맞추어 클러스터링 하는 기법이다. 공간 근접도에 따른 클러스터링 기법은 기존의 색인의 구조를 고려하지 않아 색인의 영역이 확장될 가능성이 크며, 과정이 복잡하여 클러스터링 비용이 많이 든다[6]. 하지만 Seeded 클러스터링 기법은 색인의 상위부분에 데이터를 삽입하여 클러스터링 하기 때문에 기존의 클러스터링 기법보다 노드간의 겹침이 줄어들며 빠르게 클러스터링할 수 있다[8].

2.3 색인 전송을 통한 색인 구성 기법

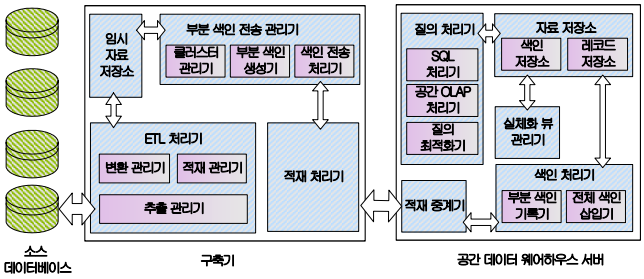
색인 전송을 통한 색인 구성 기법은 기 구축된 색인의 재활용을 위해 색인 구조를 직접 전송하여 색인을 구성한다[3]. 색인 전송으로 인한 기 구축된 색인의 물리적인 사상문제는 연속적인 익스텐트(Extent)를 예약하여 해결하였다. 이 기법은 색인 전송을 통해 색인 구성을 위한 검색, 분할, MBR 재구성 비용을 제거하여 색인 구성 비용을 줄이지만 주기적인 소스 데이터의 변경이 발생하는 공간 데이터 웨어하우스의 색인 재구성 기법으로는 적절하지 않다.

3. 부분 색인 전송을 이용한 색인 재구성 기법

본 장에서는 부분 색인 전송을 이용한 색인 재구성 기법을 제안한다. 먼저, 본 제안 기법의 환경이 되는 공간 데이터 웨어하우스 구성에 대해서 설명한 후 이를 바탕으로 구축기에서의 부분 색인 생성 기법과 부분 색인 전송 기법 및 공간 데이터 웨어하우스 서버에서 부분 색인 삽입 기법을 설명한다. 제안 기법은 다수의 트리 계열 색인에 적용 가능하며, 본 논문에서는 R-Tree를 적용한다. 또한, 본 논문에서 말하는 일반적인 삽입, 검색 방법은 기존 R-Tree의 삽입, 검색 방법과 동일하다.

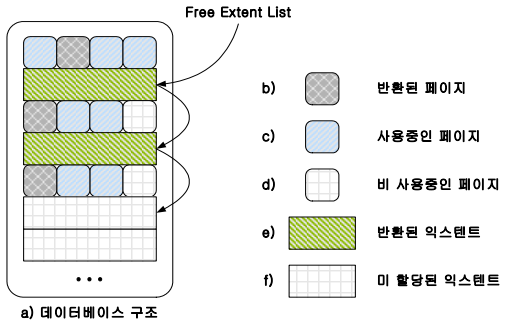
3.1 제안 기법의 공간 데이터 웨어하우스 시스템

본 기법을 적용한 공간 데이터 웨어하우스 시스템의 구성요소는 (그림 2)와 같다. 구축기는 소스 데이터베이스에서 변경된 데이터를 추출하고 변형 및 정제하며, 구축기 내부의 임시 저장소에 적재 한다. 또한, 구축기는 적재된 데이터에 대한 부분 색인을 생성하여 공간 데이터 웨어하우스 서버로 전송한다. 공간 데이터 웨어하우스 서버는 색인 처리기에서 구축기로부터 전송된 부분 색인을 기존 색인에 삽입하며, 재구성된 색인을 이용하여 공간 OLAP 연산을 지원한다.



(그림 2) 공간 데이터 웨어하우스 시스템

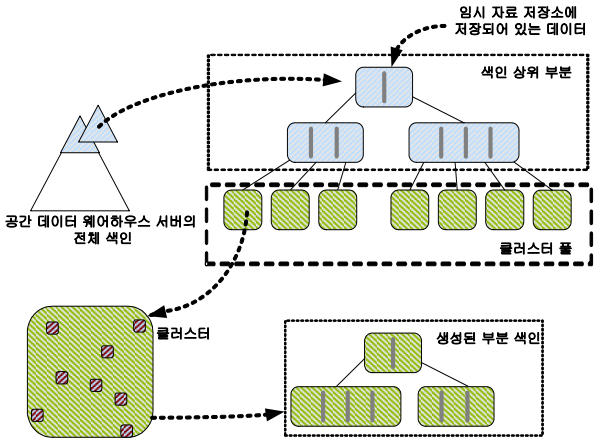
본 시스템에서의 공간 데이터 웨어하우스 서버의 자료 저장소 구조는 (그림 3)과 같으며 구축기의 임시 자료 저장소도 자료 저장소와 같은 구조를 지닌다. 본 환경에서 색인의 노드는 파일 I/O의 기본 단위인 페이지를 사용하며, 4개의 페이지는 데이터베이스 공간할당의 기본 단위인 익스텐트를 구성한다. Free Extent List는 미 할당된 익스텐트의 리스트와 데이터베이스의 테이블이나, 색인구조 등에 사용되었던 익스텐트 중 반환된 익스텐트들의 리스트를 나타낸다.



(그림 3) 자료 저장소의 구조

3.2 구축기에서 부분 색인 생성 기법

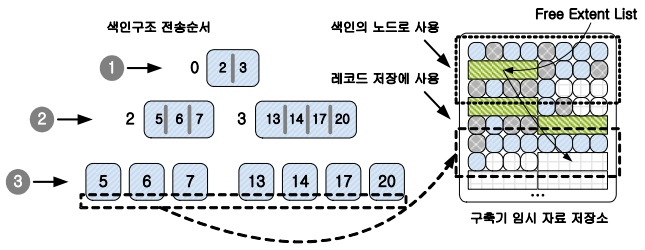
구축기는 적재되어 있는 데이터로 부분 색인을 생성하기 위해 공간 데이터 웨어하우스 서버에 있는 기 구축된 색인의 상위부분을 전송 받는다. 색인의 상위부분 레벨은 색인의 높이 및 삽입되는 데이터의 개수에 의해 결정되며, 색인의 상위부분의 레벨을 구하는 구체적인 알고리즘은 향후연구로 미룬다. (그림 4)와 같이 임시 자료 저장소에 저장되어 있는 데이터는 기존 색인의 상위 부분을 통해 클러스터링 된다. 클러스터 풀에 있는 각 클러스터는 일반적인 색인 삽입 방법으로 부분 색인을 생성한다.



(그림 4) 구축기에서 부분 색인 생성 과정

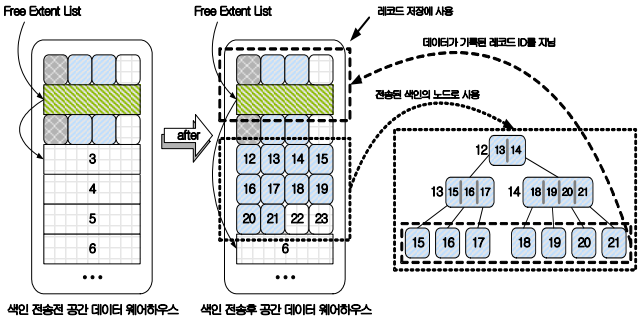
3.3 구축기에서 공간 데이터 웨어하우스 서버로 부분 색인 전송 기법

생성된 부분 색인은 (그림 5)와 같이 구축기의 임시 자료 저장소에 저장되어 있다. 사용자가 지정해 놓은 적재 시점이 되면, 구축기는 부분 색인을 위한 공간 예약 메시지를 공간 데이터 웨어하우스 서버로 전송한다. 공간 데이터 웨어하우스 서버는 부분 색인을 저장할 연속적인 익스텐트 공간을 예약한 후, 예약 완료 메시지를 구축기로 전송한다. 구축기는 예약 완료 메시지를 전송받으면 (그림 5)와 같은 순서로, 최상위노드부터 중간노드, 단말노드 순으로 부분 색인을 전송한다. 단말노드에 기록된 튜플 ID 값은 물리적 특성에 의해 공간 데이터 웨어하우스에서는 사용할 수 없기 때문에, 단말노드는 단말노드가 가리키는 해당 레코드와 함께 전송된다.



(그림 5) 부분 색인 전송

공간 데이터 웨어하우스 서버는 전송된 부분 색인을 기록한다. 하지만 전송 이전의 중간 노드 내 기록된 자식 노드 ID는 물리적 저장구조의 특성에 의해 전송 후, 공간 데이터 웨어하우스 서버의 자료 저장소에서는 사용할 수 없다. 물리적 사상 문제를 해결하기 위해 (그림 6)과 같이 공간 데이터 웨어하우스 서버는 미 할당된 연속적인 익스텐트를 필요한 색인노드 개수만큼 예약한다. 연속적으로 예약된 익스텐트는 노드들의 ID인 페이지 오프셋 값을 예측 가능하게 하고, 중간노드 내의 아이템들이 지니고 있는 자식노드 ID 값을 결정한다. 따라서 자식노드 ID 기록을 위한 부모노드로의 연속적인 차후 접근 비용은 제거된다.

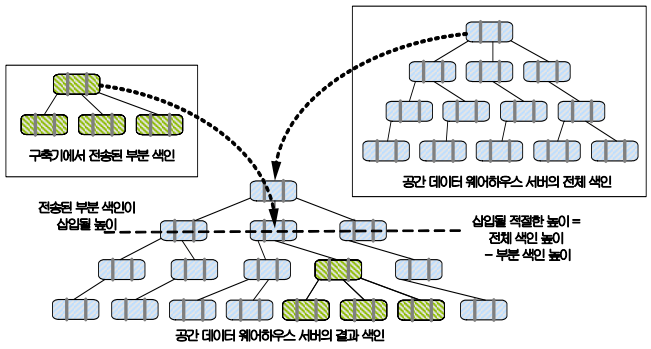


(그림 6) 전송되어 기록된 부분 색인의 구성

(그림 6)은 구축기로부터 10개의 노드로 구성된 부분 색인이 전송된다고 가정했을 시 색인 전송 전후 공간 데이터 웨어하우스 자료 저장소의 구조를 나타낸다. 공간 데이터 웨어하우스 서버는 구축기로부터 부분 색인이 전송 된다는 메시지를 받으면 미 할당된 순차적 익스텐트를 예약한다. 부분 색인을 구성하는 방법은 색인 구조 전송 순서와 같이 최상위노드부터 중간노드, 단말노드 순으로 색인을 구성한다. (그림 6)에서 12번 노드는 2개의 아이템을 가지고 있으며 각 아이템이 지닐 자식노드 ID인, 페이지 오프셋 값은 13과 14번 노드임을 예상할 수 있다. 본 기법을 이용하여 공간 데이터 웨어하우스 서버는 구축기에서 전송된 부분 색인을 다시 구성할 수 있으며, 부분 색인 저장 시 기록되어야 할 자식노드 ID를 자식노드 할당 이전에 기록함으로써 자식노드 기록을 위한 부모노드 접근 비용을 제거하였다.

3.4 공간 데이터 웨어하우스 서버에서 부분 색인 삽입 기법

공간 데이터 웨어하우스 서버는 구축기에서 전송된 부분 색인을 기존에 구축되어 있던 전체 색인에 삽입 한다. 삽입을 하기 위해 노드를 찾는 방법은 일반적인 색인의 검색 방법과 동일하다. 하지만 색인[5]의 특성상 모든 단말노드에서 높이가 같아야 하기 때문에 (그림 7)과 같이 부분 색인이 가능한 높이를 계산한다. 계산된 높이에 있는 노드 중 부분 색인이 삽입될 적절한 노드가 일반적인 검색 방법을 통해 선택된다. 선택 된 노드에 빈 엔트리가 있다면 부분 색인은 빈 엔트리에 삽입된다. 하지만 선택 된 노드에 빈 엔트리가 없다면 부분 색인은 경험적인 방법을 사용하여 삽입된다. 경험적인 방법에 관한 연구는 향후연구로 미룬다.



(그림 7) 기존 색인에 부분 색인 삽입

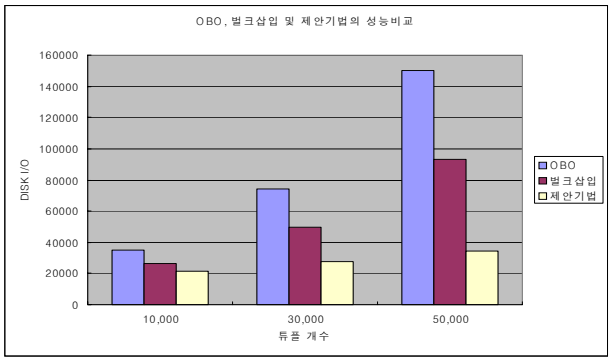
4. 성능평가

본 장에서는 일반적인 삽입 방식인 OBO(One by one)[5]와 벌크 삽입 기법[4] 및 본 논문에서 제안하는 부분 색인 전송을 이용한 색인 재구성 기법에 대한 성능평가를 하였다. 실험은 Windows 기반에서 C++를 이용하여 수행되었으며, 시스템의 구성은 공간 데이터 웨어하우스 구축기와 서버 및 소스 데이터베이스이며 속성은 <표 1>과 같다.

<표 1> 시스템 속성

시스템 속성		
공간 데이터 웨어하우스 구축기	Pentium Ⅳ 3.0GHz CPU 1G memory 120G HD	추출, 변형, 적재 담당
공간 데이터 웨어하우스 서버		공간 OLAP 연산 및 SQL 지원
소스 데이터베이스		일반 DBMS

실험에 사용할 데이터로는 표준 벤치마크 데이터인 TIGER/Line 개체 데이터를 추출하여 사용하였다[9]. 기존 색인은 100만개의 개체 데이터를 사용하여 OBO 방식으로 구축 되었으며, 클러스터를 만들기 위한 색인의 상위 부분은 기존 색인의 상위 2레벨을 이용하였다.



(그림 8) 삽입 비용 성능 비교

(그림 8)과 같이 본 논문의 제안기법은 OBO와 벌크삽입 기법보다 성능이 월등한 것을 볼 수 있다. 또한, 소스 데이터베이스에 삽입되는 데이터의 양이 많아지면 제안기법은 OBO 보다 최대 80% 삽입 성능이 좋아지며, 벌크 삽입 기법 보다 최대 45% 삽입 성능이 좋아진다.

5. 결론 및 향후연구

본 논문에서는 다량의 데이터가 주기적으로 삽입되는 환경에서 효율적인 부분 색인 전송을 이용한 색인 재구성 기법을 제안하였다. 제안 기법은 구축기에서 추출된 데이터를 색인의 구조에 맞게 클러스터링하며, 생성된 각 클러스터로 부분 색인을 생성 및 전송한다. 공간 데이터 웨어하우스 서버에서는 자식 노드 ID의 물리적 사상 문제를 해결하기 위해 물리적으로 연속된 공간을 예약하고 예약된 공간에 부분 색인을 기록한다. 기록된 부분 색인은 기존 색인에 일반적인 삽입 알고리즘으로 삽입된다. 본 기법은 색인의 전송을 통해 색인 재구성을 위한 검색, 분할, 재조정 비용을 최소로 줄일 수 있다. 제안 기법은 공간 또는 비공간 데이터의 색인 재구성 비용 절감을 위한 다수의 트리 계열 색인 구조에 적용할 수 있으며 특히, 공간 객체 삽입 시 부모 노드 엔트리의 재구성 비용이 높은 R-Tree 계열 색인의 재구성 비용 절감에 보다 효율적이다.

향후연구 과제로는 부분 색인이 삽입 되는 노드의 빈 부분을 찾는 경험적인 기법 및 기존 색인의 상위부분 레벨 계산 방법에 관한 연구가 있다. 또한, 검색 성능을 높이기 위한 클러스터링 기법 및 재 포장 기법이 있다.

참고문헌

[1] Lafond, P., “Designing and Building the Distributed Geospatial Data Warehouse Architecture”, Proceedings of The Twelfth Annual Symposium on Geographic Information Systems, 1998.

[2] S. Chaudhuri and U. Dayal, “An Overview of Data Warehousing and OLAP Technology”, ACM SIGMOD, 1997.

[3] 박상근, 김호석, 이재동, 배해영, “분산 데이터베이스 시스템에서의 색인 구성비용 절감을 위한 효율적인 색인 전송 기법”, 정보과학회, 2003.

[4] R. Choubey, L. Chen, and E. A. understeiner, “GBI: A Generalized R-tree Bulk-Insertion Strategy”, Advances in Spatial Databases, 1997.

[5] A. Guttman, “R-Trees: A Dynamic Index Structure for Spatial Searching”, Proceedings of SIGMOD, 1984.

[6] M.R. Anderberg, “Probability and Mathematical Statistics”, Academic Press, 1973.

[7] L. Chen, R. Choubey, and E. A. Rundensteiner, “Bulk-insertions into R-trees using the small-tree-large-tree approach”, ACM GIS, 1998.

[8] Taewon Lee, Bongki Moon, and Sukho Lee “Bulk Insertion for R-tree by Seeded Clustering”, DEXA, 2003.

[9] TIGER/Line Files, 2000 Technical Documentation, U.S. Bureau of Census, Washington DC, accessible via URL http://www.census.gov/geo/www/tiger/tigerua/ua_tgr2k.html.