

이동체 데이터베이스를 위한 통합 색인의 이주 정책¹

정지원⁰ 안경환 서영덕 홍봉희
부산대학교 컴퓨터공학과
{whitegem⁰, khan, ydseo, bhhong}@pusan.ac.kr

Migration Policies of a Unified Index for Moving Objects Databases

Jiwon Jung⁰ Kyoung-Hwan Ahn Young-Duk Seo Bong-Hee Hong
Dept. of Computer Engineering, Pusan Nat'l University

요 약

무선 통신 기술의 발달로 인하여 LBS(Location Based System)와 같은 새로운 이동체 관련 서비스가 생겨나고 있다. 위치 기반 서비스에서 클라이언트인 이동체들이 주기적으로 보고하는 위치 데이터를 실시간으로 처리하기 위해 서버에서는 메인 메모리 DBMS를 유지하는 것이 필요한데, 데이터의 양이 계속적으로 증가하는 특성으로 인해 메인 메모리의 공간이 부족할 때 데이터를 디스크로 옮기는 시스템 설계가 필요하다. 그러나 기존의 연구는 대용량 이동체 환경에서의 색인 이주를 위한 노드 선택 정책과 이주를 위해 선택된 노드들의 디스크 배치 정책을 통합하여 나타내지 못하였다. 그러므로 대용량 이동체 데이터베이스 시스템 환경에 적합한 이주 정책들에 대한 연구가 필요하다.

이 논문에서는 대용량 이동체 데이터베이스 환경을 고려한 노드 선택 정책과 디스크 배치 정책을 분류하고 새로운 이주 정책을 제시한다. 노드 선택 정책으로는 질의 성능을 위해서 캐쉬의 LRU(Least Recently Used) 정책을 이용한 변형된 LRU 정책을 제시하고, 삽입 우선 정책으로는 이동체 색인인 R-tree의 삽입 알고리즘을 역이용한 정책을 제시한다. 또한 이주되는 노드들에 대한 디스크 페이지 배치가 시스템의 질의 성능에 영향을 미치므로 이를 고려한 디스크 배치 정책을 제시한다.

1. 서 론

이동 통신 기술의 발달로 인하여 무선 이동 기기의 사용이 보편화되면서 LBS(Location Based System)의 요구가 날이 갈수록 증가되고 있다. 이와 같은 위치 기반 서비스에서 클라이언트인 이동체들은 서버로 주기적으로 위치 데이터를 전송한다. 이런 대용량 데이터의 빈번한 갱신과 빠른 질의 처리를 위해서 서버에서는 메인 메모리 DBMS를 사용한다. 그러나 이동체 데이터의 양은 계속적으로 증가하는 특성을 가지므로, 메인 메모리의 공간 부족이 발생할 때 데이터를 디스크로 옮기는 시스템 설계가 요구된다.

기존의 연구는 대용량 이동체 환경에서의 색인 이주를 위한 노드 선택 정책과 이주를 위해 선택된 노드들의 디스크 배치 정책을 통합하여 다루지 않았다. 즉, 보고되는 위치 데이터가 누적되어 저장할 메모리 공간이 부족할 경우, 디스크로의 이주를 위한 후보를 고르는 노드 선택 정책이 필요하다. 또한 질의 시 디스크로 옮겨진 노드들에 대한 참조가 일어날 수 있는데, 디스크로부터의 참조는 시스템 전체 성능에 영향을 미치므로 이를 위한 디스크 배치 정책이 필요하다. 그러므로 대용량 이동체 데이터베이스 시스템 환경에 적합한 이주 정책에 대한 연구가 필요하다.

이 논문에서는 대용량 이동체 데이터베이스 환경을 고려한 노드 선택 정책과 디스크 배치 정책을 분류하고 새로운 이주 정책을 제시한다. 노드 선택의 주체에 따라 관리자와 시스템에 의한 선택으로 분류하고, 색인의 목적에 따라 질의 우선 정책과 삽입 우선 정책으로 분류하였다. 질의의 성능을 위해서 캐쉬의 LRU 정책을 이용한 변형된 LRU 정책을 제시하고, 삽입 우선 정책으로는 이동체 색인인 R-tree의 삽입 알고리즘을 역이용한 정책을 제시한다. 더불어 이주되는 노드들에 대한 디스크 페이지 배치가

이동체 데이터베이스 시스템의 질의 성능에 영향을 미치므로 이를 위한 디스크 배치 정책을 제시한다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구들에 대해서 알아보고, 3장에서는 대상 환경 및 문제 정의를 기술한다. 4장에서는 이주 정책을 분류, 설명하고 새로운 정책을 제안한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해 설명한다.

2. 관련 연구

기존에는 데이터의 양, 사용 빈도에 따라 데이터의 일부분을 다른 저장 매체로 이주시키는 것에 관한 연구[1, 2]와 정적 데이터를 디스크 상에서 클러스터링(clustering)하는 연구[3]이 있었다.

데이터를 다른 저장 매체로 이주시키는 것에 관한 연구로는 [1, 2]가 있다. [1, 2]에서 데이터는 현재 데이터와 과거 데이터 두 부분으로 나뉜다고 가정한다. 이 연구들은 현재 데이터는 자기 디스크에 저장하고 시간이 지난 과거 데이터는 광학 저장 장치로 이주하는 방법에 관한 연구이다. 그러나 [1, 2]는 단순히 과거 데이터를 광학 저장 장치로 옮길 뿐 질의 시에 빠른 접근을 고려한 데이터 클러스터링에 대한 연구는 없었다.

정적 데이터를 디스크 상에서 클러스터링하는 연구로는 [3]이 있다. 이 연구들은 데이터 접근 시 원하는 페이지로의 디스크 헤드 접근 시간이 페이지당 전송 시간보다 훨씬 크기 때문에 디스크 I/O 시간을 줄이기 위해 데이터를 적절히 배치하는 연구이다. 그러나 어느 연구도 실시간 대용량 이동체 환경에서의 과거 데이터를 메인 메모리에서 디스크로 이주시키기 위한 데이터 배치에 관한 연구는 없었다.

¹ 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음.
(KRF-2003-002-D00278)

3. 대상 환경 및 문제 정의

3.1 대상 환경

이 논문에서는 이동체가 주기적으로 서버에 자신의 위치 정보와 필요시 질의 정보를 전송한다고 가정한다. 서버에서는 위치 정보를 메인 메모리에 저장하고, 메모리 공간이 부족할 경우 과거 이력 데이터는 디스크로 옮겨진다. 클라이언트에서 질의 요청이 오면, 저장하고 있는 위치 정보를 이용하여 응답하는 환경을 대상으로 하고 있다. 이 논문에서는 이동체 데이터베이스 환경에서의 데이터 이주 정책들을 분류하고 제안한다.

3.2 문제 정의

클라이언트인 이동체들이 주기적으로 서버에 위치 데이터를 보고할 때, 데이터는 메인 메모리 색인에 계속 쌓여가게 된다. 계속된 삽입으로 인해서 메모리 공간이 부족할 경우에 과거 이력 데이터들은 디스크로 옮겨질 필요가 있다. 이때 고려해야 할 것은 첫째로 어떤 노드들을 선택할 것인지, 둘째로 선택된 노드들을 어떻게 디스크 상에 배치할 것인가이다.

첫째, 노드 선택(Nodes Selection)은 데이터 삽입, 질의 등에 영향을 미친다. 기존에는 노드 생성 시간(Node Creation Time)을 기준으로 이주시킬 노드를 선택했다. 색인의 노드가 생성된 시간에 따라 차례대로 선택하는 방법으로서 기존의 디스크 기반 색인의 디스크 배치 순서와 동일하다. 선택 알고리즘은 간단하지만 삽입 및 질의에 대한 고려가 부족하다. 즉 그림 1을 고려해 볼 때 다음과 같은 문제가 발생할 수 있다[4].

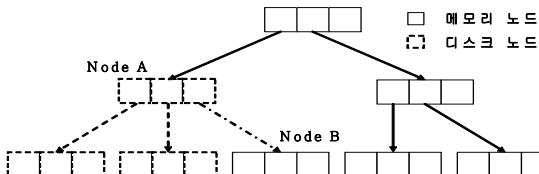


그림 1. 노드 생성 시간에 따른 노드 선택 정책의 문제점

먼저 노드B가 디스크로 이주되어야 하는 경우 부모 노드의 포인터 갱신을 위해서 노드A를 다시 메모리로 읽어 들여야 한다. 또한 노드B가 질의에 의해 참조된다면 노드A가 먼저 메모리로 읽혀져야 한다. 이와 같이 잘못된 노드 선택에 의해서 디스크 I/O가 발생하는 것은 바람직하지 못하다. 따라서 노드 이주 시에 그림 1과 같은 상황이 발생하지 않도록 노드 선택 정책을 정할 필요가 있다.

둘째, 디스크 배치(Disk Placement)는 질의 성능에 영향을 미친다. 그림 2는 R-tree[5]가 생성되는 과정에서 각 노드들이 디스크 상에 어떠한 순서로 배치되는 것인가를 나타낸 것이다.

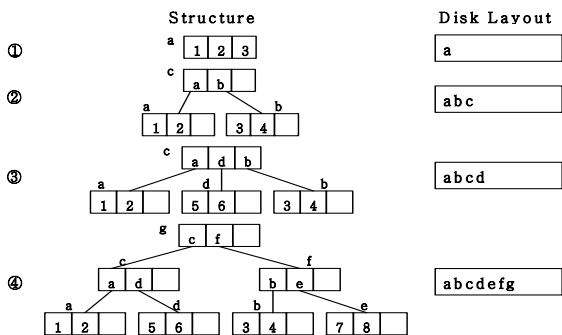


그림 2. 기존 방식에서의 디스크 구조

일반적으로 색인이 구성될 때는 그림 2와 같이 노드가 생성된

순서대로 페이지들이 디스크 상에 할당된다. 그런데 질의 시에 색인의 단말 노드까지 찾기 위해서 탐색 시간이 증가하게 되어, 전체 디스크 I/O 시간의 증가를 가져온다. 이는 시스템에 있어 매우 중요한 전체 용량에 영향을 미치게 된다.

빠른 응답이 필요한 이동체 데이터베이스 시스템에서는 이와 같은 문제를 지양해야 한다. 따라서 이를 위한 디스크의 특징과 질의 성능을 고려한 디스크 배치 정책이 필요하다.

4. 이주 정책(Migration Policy)

3장에서 언급한 문제를 해결하기 위해서 4장에서는 노드 선택 정책과 디스크 배치 정책을 통합한 색인 이주 정책을 제안한다. 그림 3은 제안하는 색인 이주 정책의 일련의 수행과정을 간략히 그림으로 도식한 것이다.

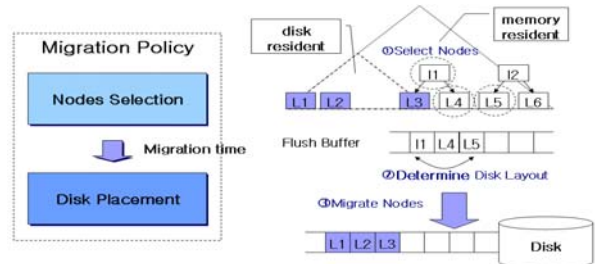


그림 3. 이주 정책(Migration Policy)

색인의 이주 순서는 다음과 같다. 먼저, 이동체 위치 정보의 계속된 삽입으로 인해서 메모리 용량이 한계에 다다르면, 과거 데이터에 해당하는 부분을 노드 선택 정책에 따라 고른다. 다음으로 디스크 배치 정책에 따라 선택된 과거 데이터를 질의 시, 접근이 용이하도록 버퍼 상에서 적절히 배치한다. 마지막으로 버퍼의 내용을 디스크로 쓴다.

4.1 노드 선택 정책 (NSP : Nodes Selection Policy)

기존 연구는 노드 선택에 대한 기준과 사용 목적에 맞는 정책들을 제시하지 않았다. 따라서 이 논문에서는 노드 선택의 주체에 따라, 색인이 사용되는 목적에 따라, 노드 선택 기준을 분류하고 노드 선택 정책들을 제안한다.

노드 선택 주체에 따른 분류

먼저 시스템 관리자가 이주시킬 노드를 선택한다면 관리자는 단순히 이주 조건을 시스템에 설정하면 된다. 예를 들면 `현재로부터 30일간의 궤적 데이터만 메모리에 적재하라`와 같은 조건을 시스템에 설정하는 것이다.

다음으로 이주시킬 노드 선택이 시스템에 의해서 자동적으로 수행되는 경우이다. 이 때는 노드 선택이 메모리의 여유 공간 및 이동체의 특징에 따라 시스템의 판단에 따라 자동적으로 관리된다. 이 논문에서 다루는 분야는 후자이다.

색인의 목적에 따른 분류

색인이 사용되는 응용의 목적에 따라 질의 우선 정책, 삽입 우선 정책으로 분류할 수 있다.

첫째, 질의 우선 정책은 질의 시에 자주 사용될 것 같은 노드는 메인 메모리 상에 유지될 수 있도록 선택하는 것이다. 둘째, 삽입 우선 정책은 삽입 시에 사용될 것 같은 노드는 메인 메모리 상에 유지될 수 있도록 선택하는 것이다. 다음은 제안하는 노드 선택 정책이다.

NSP 1 : 변형된 LRU 정책 (질의 우선 정책)

캐시의 LRU(Least Recently Used) 정책과 비슷하게 가장 참조된지 오래된 노드를 선택하는 것이다. 그러나 기존의 방식과 유사하지만 약간의 수정이 필요하다.

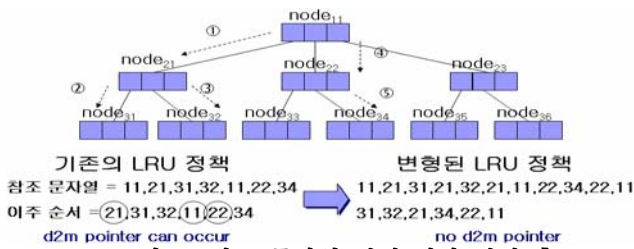


그림 4. 참조 문자열 변경 전과 변경 후

그림 4의 노드들이 ①~⑤ 순서로 참조된다고 가정할 때, 노드 참조 순서를 나타낸 참조 문자열의 노드 번호는 중복되지 않는다. 따라서 구성된 참조 문자열에 대해 LRU 정책을 적용하여, 이주 순서를 정하고 해당 노드들을 이주시키게 된다. 이 때 21, 11, 22번 노드와 같이 자식 노드가 이주되지 않은 상태에서 부모 노드가 먼저 이주되는 경우가 생긴다. 즉, d2m pointer(disk to memory pointer)[4]가 발생하여, 이주되지 않은 자식 노드를 참조하기 위해서 불필요한 디스크 I/O가 필요하게 된다. 이는 참조 문자열을 구성할 때, 참조되는 노드들을 순서대로 중복해서 저장함으로써 해결할 수 있다. 변경된 참조 문자열에 대해 LRU 정책을 적용함으로써 노드 이주 순서를 달리하여 d2m pointer 발생을 제거하였다.

변형된 LRU 정책은 기존의 교체 전략을 활용함으로써 구현이 쉬우며, 질의 처리 시에 효율적이라는 장점을 가진다. 반면에 선택된 노드들 간에 논리적 연관성이 작아서 디스크 배치 정책에 불리하다.

NSP 2 : 삽입 알고리즘을 역이용한 정책 (삽입 우선 정책)

삽입 알고리즘에 의해서 사용되지 않을 것 같은 노드를 선택하는 것이다. 여기서 중요한 점은 삽입에 사용되는 노드는 디스크에 존재해서는 안된다는 것이다. 만약 클라이언트로부터 보고가 들어왔을 때, 삽입에 사용될 노드가 디스크에 있다면 디스크 접근 시간 때문에 처리 시간이 지연된다. 따라서 다음 보고가 들어왔을 때 처리를 못하는 상황이 계속해서 발생한다.

R-tree[5]의 ChooseSubtree 알고리즘은 엔트리 삽입을 위해서 노드 확장이 최소로 이뤄지는 노드를 선택하는 최소 영역 확장 정책을 사용한다. 이에 착안하여 이주시킬 노드를 선택하기 위해서 최소 영역 확장 정책과는 반대로 노드 확장이 최대로 이뤄지는 노드를 선택하는 최대 영역 확장 정책을 사용한다. 즉, 삽입될 확률이 가장 낮은 노드를 이주시키는 정책이다.

삽입 알고리즘 역이용 정책은 기존의 알고리즘을 활용할 수 있고 대용량 이동체 데이터 삽입에 효율적이지만, 질의에는 비효율적일 수 있다.

4.2 디스크 배치 정책 (DPP : Disk Placement Policy)

디스크 배치는 메인 메모리 성능을 고려한 노드 선택 정책에 의해서 선정된 노드들을 디스크 상에 어떻게 배치하는가를 결정하는 것이다. 일반적으로 디스크에서의 탐색 시간은 전송 시간보다 수 배 이상 길다. 그러므로 이 논문의 디스크 배치 목표는 질의에 의해 노드가 참조될 때, 디스크 탐색 시간을 줄이는 것이다. 다음은 제안하는 디스크 배치 정책이다.

DPP : 트리 레벨 동적 클러스터링

[6]은 디스크 상의 페이지들이 서로 가까이 위치한다면 디스크 I/O 속도는 더욱 빨라진다고 하였다. 그림 5는 노드 선택 정책에 의해서 선택된 전체 트리의 일부분을 나타내고, 이 노드들이 디스크 상에 어떻게 나열되는가를 보여준다.



그림 5. 동적 클러스터링을 적용한 후의 디스크 구조

그림 5의 A는 왼쪽의 선택된 노드들을 3장 문제 정의에서 언

급한 노드 생성 시간 순서로 기록했을 때의 디스크 배치이다. 이 논문에서 제안하는 방법은 그림 5의 B와 같이 레벨 별로 모아서 디스크에 기록하는 트리 레벨 동적 클러스터링이다. 즉, 루트 노드부터 단말 노드까지의 각 레벨 별로 노드들을 디스크에 순차적으로 기록하는 것이다.

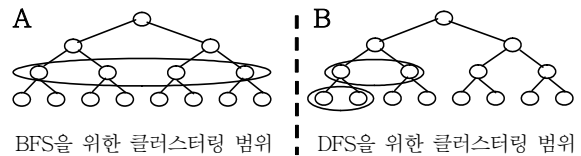
여기서 탐색 시간(seek time) t_s 은 9 msec, 페이지 당 전송 시간 t_t 은 1 msec 그리고 buffer의 크기는 5라 하자. 빈 페이지들의 시퀀스를 gap[6]이라 할 때, 읽혀질 빈 페이지들의 최대 시퀀스를 2라고 가정한다. 다음은 그림에 표시된 질의 영역에 대해 질의를 수행했을 때, A, B 각각의 디스크 배치 상황에서 참조된 노드들에 대한 총 디스크 I/O 시간이다.

$$A: (1t_s + 1t_t) + (2t_s + 2t_t) + (3t_s + 5t_t) = 6t_s + 8t_t = 62 \text{ msec}$$

$$B: (1t_s + 1t_t) + (1t_s + 2t_t) + (1t_s + 7t_t) = 3t_s + 10t_t = 37 \text{ msec}$$

B의 디스크 I/O 시간은 A보다 약 절반 가량 줄었음을 볼 수 있다. 이는 제안하는 클러스터링 정책을 적용함으로써, 이동체 데이터베이스 시스템의 속도 향상을 꾀할 수 있음을 의미한다.

그런데 색인의 검색 방법에는 BFS(Breadth First Search)와 DFS(Depth First Search)가 있다. 이러한 검색 방법에 따라 클러스터링 적용 범위를 다르게 할 수 있다. 그림 6은 BFS와 DFS에 따른 클러스터링의 적용 범위를 나타낸 것이다.



BFS를 위한 클러스터링 범위 DFS를 위한 클러스터링 범위

그림 6. BFS와 DFS에 따른 클러스터링 적용 범위

먼저 BFS의 경우, 그림 6의 A처럼 루트에서 단말 레벨까지 각 레벨 별로, 같은 레벨에 속한 노드들을 순서대로 디스크에 기록하는 방법을 사용한다. 반면에 DFS의 경우, 재귀호출을 통해 탐색하는 특징을 가지므로, 그림 6의 B처럼 같은 부모 노드를 가진 형제 노드 단위로 디스크에 기록하는 방법을 사용한다.

5. 결론 및 향후 연구

이 논문에서는 이동체 데이터베이스 환경에서 빈번하게 보고되는 대용량의 이동체 위치 데이터를 효율적으로 처리하기 위한 색인 이주 정책을 제안하였다. 이를 위해 노드 선택 정책과 디스크 배치 정책으로 분류하고 새로운 정책을 제안하였다.

노드 선택의 주체와 색인이 사용되는 목적에 따라 노드 선택 기준을 분류하고 질의 우선, 삽입 우선 정책에 따른 노드 선택 정책들을 제시하였다. 또한 디스크 페이지의 배치가 실시간 시스템의 질의 성능에 영향을 미치므로 이를 위한 디스크 배치 정책을 제시하였다.

향후 연구로는 노드 이주 시점과 이주 노드의 크기에 대한 분석이 필요하다. 더불어, 이 논문에서 제시한 정책들에 대한 구현 및 성능 평가를 통한 검증이 있어야 할 것이다.

참고 문헌

- [1]C. Kolovson and M. Stonebraker, "Indexing techniques for historical databases," Fifth IEEE International Conference on Data Engineering, pp 127-137, 1989.
- [2]D. Lomet and B. Salzberg, "Access Methods for Multiversion Data," Proc. of SIGMOD, pp 315-324, June 1989.
- [3]D.S. Cho, B.H. Hong, "Optimal Page Ordering for Region Queries in Static Spatial Databases," 11th International Conference DEXA, pp 366-375, 2000.
- [4]K.H. Ahn, B.H. Hong, "이동체 데이터베이스를 위한 통합 색인 기법," 한국정보과학회 데이터베이스 연구(KDBC 2003) 19권 2호 pp 36-43, 2003년 5월.
- [5]A. Guttman, "R-trees: A dynamic index structure for spatial searching," ACM SIGMOD Conference, pp 47-54, 1984.
- [6]B. Seeger, P.A. Larson, and R. McFayden, "Reading a set of disk pages," Proc. 19th Int. Conf. on VLDB, pp 592-603, Dublin, Ireland, August 1993.