

# LBS 플랫폼에서 실시간 서비스를 위한 하이브리드 저장 관리자 설계

°박준용\*, 김호석\*, 김명근\*, 배해영\*  
\*인하대학교 컴퓨터·정보공학과  
e-mail:afetra@dblabb.inha.ac.kr

## Design of Hybrid Storage Manager for Realtime Services in LBS Platform

°Jun-Yong Park\*, Ho-Seok Kim\*, Myoung-Keun Kim\*,  
Hae-Young Bae\*

\*Dept. of Computer Science & Engineering, Inha University

### 요 약

위치기반 서비스 시스템에서 주 요구사항은 사용자의 실시간 위치 정보 검색과 질의에 대한 빠른 응답시간이다. 그러나 기존의 위치기반서비스 시스템에서 저장장치로 사용하는 디스크 기반 데이터베이스 시스템은 빈번한 디스크 I/O로 인해 발생하는 질의응답에 대한 시간지연으로 이러한 요구사항을 만족시키지 못한다. 또한 주기억장치 데이터베이스 시스템은 한번에 주기억장치에 적재할 수 있는 양이 제한되어 있기 때문에 대용량의 공간 데이터를 처리하는데 문제가 있다. 본 논문에서는 위치기반서비스를 위하여 위치 정보 데이터의 다단계 저장 관리를 통해 LBS 플랫폼에서 실시간 서비스를 지원하는 하이브리드 저장 관리자를 제안한다.

하이브리드 저장 관리 시스템은 위치기반 서비스 시스템에서 대용량의 이동체 데이터의 효과적인 저장 및 관리를 위해 주기억장치 데이터베이스 관리시스템과 디스크 기반의 공간 데이터베이스 관리시스템을 통합한 구조를 가진다. LBS 플랫폼에서 빠른 응답을 필요로 하고 자주 사용되는 데이터는 메모리 기반의 데이터베이스에 관리를 하고, 사용 빈도가 적은 데이터는 디스크 기반의 데이터베이스에 관리하는 하이브리드 저장 관리자를 제시한다.

### 1. 서론

최근 개인 생산성 향상(Personal Productivity)을 위한 응용분야인 위치기반서비스(Location Based Service)는 실시간 환경에 적합한 빠른 응답시간을 요구한다. 하지만 기존의 디스크 기반 데이터베이스는 모든 데이터를 디스크에 저장, 관리하기 때문에 데이터를 다루기 위한 디스크 접근의 오버헤드가 너무 커서 빠른 응답시간을 요구하는 실시간 응용환경에 적합하지 않다. 또한 주기억장치 상주 데이터베이스는 주기억장치에 영구적인 데이터베이스를 구축, 관리함으로써 데이터 접근에 대한 오버헤드를 낮추고 빠른 응답성을 보여주는 장점을 갖고 있지만, 메모리공간의 제한성과 메모리의 휘발성 같은 제한성을 시스템이 갖고 있다[1,4].

본 논문에서는 이러한 제약조건을 극복하고 실시간 서비스를 지원하기 위한 하이브리드 저장 관리

자를 제안한다. 본 논문에서 제안하는 하이브리드 저장 관리자는 주기억장치 기반의 데이터베이스와 디스크 기반의 데이터베이스를 통합 관리하여 대용량 공간 데이터 처리의 빠른 응답 시간을 보장하고 LBS 플랫폼에서 실시간 서비스를 가능하게 한다.

이 논문의 구성은 다음과 같다. 2장에서는 LBS 플랫폼에서 실시간 서비스 지원을 위한 저장 구조와 다중 레벨 저장 관리에 대한 내용을 간략하게 소개하고, 3장에서는 본 논문에서 제시하는 실시간 서비스 지원을 위한 하이브리드 저장 관리 구조에 대해 설명을 한다. 끝으로 4장에서는 결론 및 향후 연구 방향을 제시한다.

1) 본 연구는 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

2. 관련연구

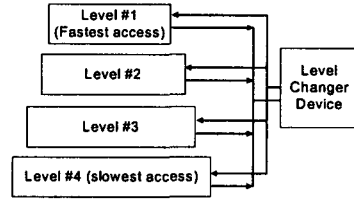
2.1 위치 기반 서비스에서의 저장 구조

위치기반서비스를 지원하기 위한 기술로서 대표적인 것으로 사용자 위치를 지도상에서 표현하기 위한 지도서비스 컴포넌트, 경로 탐색(Routing) 및 트래킹(Tracking) 컴포넌트, 현재 위치에서 주어진 영역내에 위치한 장소를 서비스하는 디렉토리(Directory) 서비스 컴포넌트(디렉토리 서비스는 POI:Point Of Interest 또는 AOI:Area of Interest 서비스로도 불린다), 세일 등 광고를 특정 위치에 위치한 모든 사용자에게 서비스하는 Event Notification 서비스 등이 있으며, 이의 종류는 LBS의 응용에 따라서 확장 가능하다. 위치데이터 서버는 LBS가 가지는 이동성을 만족시키기 위하여, 이동하고 있는 사용자 객체의 정보를 획득, 저장, 검색, 갱신하는 기술로서 일반적으로 대용량의 정보를 실시간으로 처리하기 위한 데이터베이스 기술과 인덱싱 기술들의 활용이 필요하다. 특히 이러한 실시간 처리를 위해 이동 객체의 위치 정보가 데이터베이스에 저장 및 처리되는 과정은 현재 위치 테이블(Current Location Table)을 별도로 관리하기 때문에 과거 위치 테이블(Historical Location Table)과 더불어 2개의 테이블을 유지 및 관리한다. 기존 데이터 관리시 분산 환경의 시스템으로 주기억장치와 디스크에 데이터를 분할 관리를 통해 실질적인 실시간 처리를 하기에 네트워크 환경과 같은 외부적인 요인에 영향을 받는 문제를 포함하고 있는데, 본 논문에서는 자주 사용되는 현재 위치 테이블과 같은 자주 사용되어지는 데이터에 대해 구분해서 두가지 색인 방법을 적용한다. 첫째, 주기억장치에서는 공간객체에 대한 빠른 검색을 지원하는 버전 기반의 R-tree 색인을 사용한다. 버전 기반의 R-tree 색인은 논리적 버전과 물리적 버전을 이용하여 검색연산과 갱신연산이 서로 다른 버전에서 실행되게 함으로써, 검색연산이 갱신연산과의 직접적인 충돌을 회피하여 갱신연산으로 인한 검색연산의 블록킹을 제거하여 이동 객체에 대한 질의에 빠른 처리를 제공한다. 둘째, 디스크 기반의 장치에서는 사용 빈도가 적은 과거 위치 테이블 같은 데이터를 보관하기 위해 일반적인 R-tree 색인을 구성하여 관리를 한다. [2,4,6].

2.2 다중 레벨 저장 관리

첨차 디스크 또는 주기억장치 기반의 데이터베이스

스에서 다루는 데이터의 크기가 커지고 있다. 따라서 다양한 데이터 크기 처리를 위한 저장 구조가 필요하게 되었고, 이에 상이한 저장 장치들을 다중 레벨 저장 구조로서 데이터를 관리하는 기법이 제안되었다.



[그림2] 다중 레벨 저장 관리

[그림1]은 다중 레벨의 저장장치를 Level Changer Device를 이용하여 데이터를 저장하고 레벨에 따라 관리하는 것을 나타낸 그림이다. 저장 시스템의 모든 데이터 흐름은 Level Changer Device에 의해 통제된다. 데이터의 흐름은 위치적으로 인접한 레벨로 바로 전송되지 않고, 또한 임의의 레벨로도 전송되지 않는다. 단지 Level Changer Device를 통해 모든 데이터 전송 경로가 결정되고 데이터의 전송을 결정한다[3].

Location	Main Memory Database	Disk Database	Archive Database
Main Memory	Main Memory objects in Main Memory format	Cache of disk blocks	Cache of archive blocks
Disk		Disk objects in Disk format	Cache of archive blocks
Archive			Archive objects in Archive format

[그림2] 세개의 저장 레벨에서의 논리적 모델

다중 레벨에서 저장하고 있는 중요 데이터에 대한 관리 방법은 다음과 같다. 빠르게 수행하려는 데이터에 대해서는 빠른 처리를 수행할 수 있는 레벨의 저장 장치로 데이터를 위치시켜 처리를 수행한다. 아카이브 파일에 존재하는 중요한 블록에 대해서는 디스크 블록에 저장하고, 디스크에 존재하는 중요한 블록은 주기억장치에 위치하여 관리한다. [그림2]는 세 개의 저장 레벨에서 처리하는 논리적인 모델을 나타낸 그림이다[5]. 본 논문에서는 하나의 시스템에서 다중 레벨 저장 관리기법을 활용하여 디스크 기반과 주기억장치 기반의 데이터베이스를 통합하여 사용하는 하이브리드 저장 관리자를 제안한다.

3. 하이브리드 다중 레벨 저장 관리자

이 장에서는 LBS 플랫폼에서 실시간 서비스를 지원하기 위해 본 논문에서 제안하는 하이브리드 저장 관리자를 설명한다. 빠른 서비스를 위해 어떻게

이동 객체가 저장되고 위치 정보 서비스가 처리되는 지 기술한다.

### 3.1 하이브리드 질의 처리기

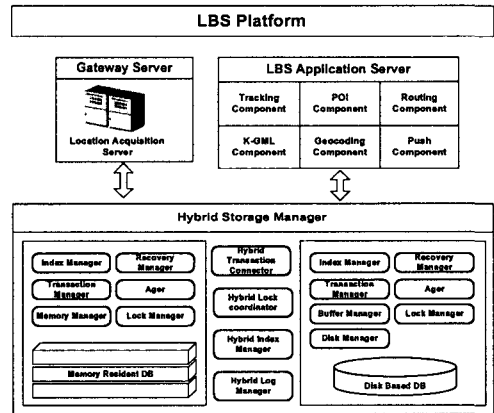
하이브리드 저장 관리자는 주기억장치 데이터베이스 관리시스템과 디스크 기반의 공간 데이터베이스 관리시스템을 통합하여 대용량 공간 데이터 처리의 빠른 응답 시간을 보장하고, 실시간 대용량 이동체 데이터의 효과적인 저장 및 관리를 위해 개발된 저장 관리자이다. 디스크와 주기억장치와 같은 다중 레벨을 관리하기 위해서는 단일 저장 관리자에서 사용하던 질의 처리기와는 다른 다중 레벨 저장 구조를 관리하기 위한 질의 처리기가 필요하다. 따라서 본 논문에서는 다중 레벨 저장 구조를 관리하기 위해 하이브리드 질의 처리기를 사용한다. 하이브리드 질의 처리기는 메타정보로 각 테이블에 데이터가 디스크 기반에만 존재하는지 아니면 디스크 기반과 주기억장치에 데이터가 저장되는지에 대한 존재 여부를 관리한다. 만약 디스크에만 존재하는 데이터를 가지고 질의 처리하는 경우 하이브리드 질의 처리기는 디스크의 저장 관리자를 사용하여 질의를 처리하고, 디스크와 주기억장치에 모두 존재하는 데이터에 대한 질의 처리인 경우 하이브리드 질의 처리기는 각각의 저장 관리자를 호출하여 처리하는 방식을 취한다.

### 3.2 하이브리드 저장 관리 구조

LBS 플랫폼에 하이브리드 저장 관리자를 사용하여 서비스를 제공하는 구조는 [그림3]과 같다. 하이브리드 저장 관리자는 실시간 처리를 위해 자주 사용되고, 빠른 응답을 요구하는 데이터는 주기억장치 기반의 데이터베이스에 저장하고, 사용 빈도가 적은 데이터는 디스크 기반의 데이터베이스에 분리 저장한다.

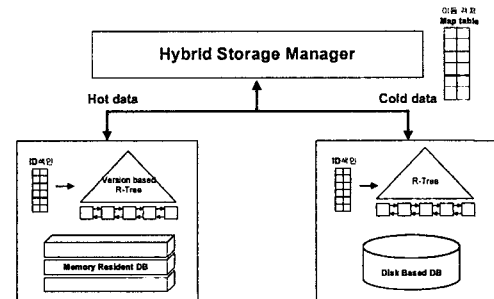
위치 획득 서버로부터 이동 객체에 대한 정보를 획득하게 되고, 이동 객체의 위치 정보는 위치 보고 시점의 시간 정보와 해당 위치 좌표로서 저장된다. 이동 객체에 대한 정보를 주기억 장치와 디스크 기반에 저장하여 사용하는데, LBS 플랫폼에서 원하는 서비스를 실시간 처리를 위해 주기억 장치에 저장되는 이동 객체는 주기억장치에서 공간객체에 대한 빠른 검색을 지원하는 버전 기반의 R-tree 색인을 사용한다. 버전 기반의 R-tree는 기존 R-tree 색인에 의한 검색연산이 검색할 노드(Node)에 공유 잠금

(Shared Lock)을 선점함으로써 같은 노드에 대한 갱신연산을 방지하지만 이러한 잠금 기반의 제어 방법은 검색연산이 갱신연산의 잠금으로 인하여 오랫동안 지연되는 블록킹 오버헤드(Blocking Overhead)를 갖게되어 제안된 방법이다. 버전 기반의 R-tree를 사용함으로써 갱신연산을 위하여 논리적 버전과 물리적 버전을 생성함으로써, 검색연산과의 직접적인 충돌을 피한다. 그래서 검색연산은 노드 탐색을 위하여 어떤 잠금이나 래치(Latch)의 선점이 불필요하므로 검색연산은 갱신연산으로 인한 지연이 발생하지 않는 빠른 처리를 제공한다. 그리고 디스크 기반의 데이터베이스에 저장되는 데이터는 일반적인 R-tree 색인을 구성하여 관리한다.



[그림 3] LBS 플랫폼에서의 서비스 처리 구조

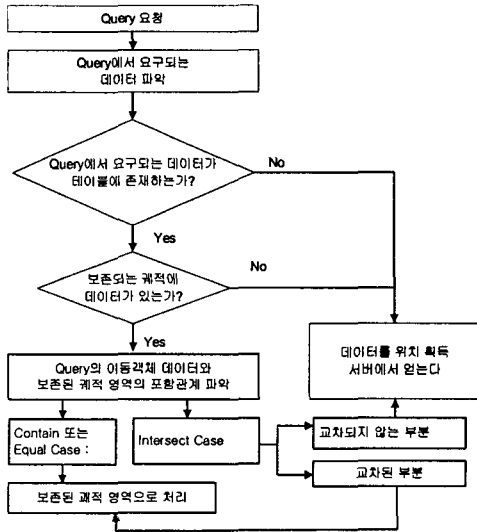
주기억장치와 디스크에 이동객체에 대한 색인을 구성 할때, 이동 객체가 어느 위치에 존재하는지 파악하기 위해 이동 객체의 전체 맵 테이블을 생성하여 이동 객체의 위치를 파악하고자 할때 사용한다. 이와 같이 다중 레벨 저장 구조를 사용한 하이브리드 저장 관리 구조는 [그림4]와 같다.



[그림 4] 하이브리드 저장 관리

### 3.3 위치 정보 서비스 방법

이동 객체 정보의 관리는 위치 획득 서버로부터 데이터를 획득하고, 획득한 정보에 대해 LBS 서비스의 요청에 대해 제공하게 되는데 서비스 과정은 [그림5]와 같다.



[그림5] 이동객체 데이터 요구 처리 순서도

LBS 플랫폼에서 서비스를 제공하기 위해 임의의 이동 객체 데이터를 요청할때, 하이브리드 저장관리자에 의해 이동 객체의 위치 정보가 존재하는지 파악한다. 주기억장치에 요구하는 위치 정보가 존재한다면 해당 데이터에 대한 위치정보를 버전 기반의 R-tree에서 유지하는 영역과 비교하여 데이터를 처리하고, 그 데이터의 보관을 위해 우선순위를 높게 설정한다. 만약 주기억장치에 존재하지 않고 디스크에 존재한다면 그 위치 정보를 디스크에서 처리하고, 주기억장치에서 관리하는 테이블에 현재 처리한 데이터의 위치 정보를 추가하여 관리하게 한다. 만약 디스크에도 위치 정보가 관리되고 있지 않을 경우에는 위치 획득 서버로부터 위치를 획득하여 처리를 한다.

### 4. 결론 및 향후 연구 방향

본 논문에서는 디스크 장치 기반과 주기억장치 기반을 통합하여 관리하는 하이브리드 저장 관리자를 제안하여 LBS 플랫폼에서의 이동객체 데이터에 대한 실시간 서비스 지원하기 위한 방안을 제시하였

다. 하이브리드 저장 관리를 통해 자주 사용하는 데이터는 메모리 기반 장치에 관리를 하고 사용도가 적은 데이터에 대해서는 디스크 관리를 하여 디스크 기반의 문제점인 데이터 접근에 따른 I/O비용 해결과 주기억장치의 내제적인 문제점인 제한된 메모리에 대한 방안을 제시하였다.

향후 연구로는 본 논문에서 제안한 하이브리드 저장 관리자에 대한 성능 테스트로 실시간 서비스를 위한 시스템의 증명이 필요하다. 그리고 하이브리드 저장 관리에서 실시간 서비스를 제공하기 위한 최적화를 위한 연구 개발이 필요하다.

### 참고문헌

- [1] A. Ammann, M. Hanrahan and R. Krishnamurthy, "Design of a Memory Resident DBMS", Proceedings of a IEEE COMPCON, pp.54-57, 1985.
- [2] D. Pfoser, Christian S. Jensen, Yannis Theodoridis, "Novel Approaches to the Indexing of Moving Objects", In Proc. of the 26th VLDB Conf, 2000.
- [3] Edward Morenoff and John B. McLean, "Application of Level Changing to a Multilevel Storage Organization", Communications of the ACM Volume 10 , Issue 3 pp 149-154, 1967.
- [4] L. Forlizzi, R. H. Guting, E. Nardelli, and M. Schneider, "A Data Model and Data Structures for Moving Objects Databases", In Proc. of the ACM SIGMOD pp. 319-330 Conf., 2000.
- [5] Michael Stonebraker "Managing Persistent Objects in A Multi-Level Store" SIGMOD Conference, pp2-11, 1991.
- [6] 김민성 "공간 데이터베이스 관리 시스템에서 논-블록킹 검색을 위한 버전 기반 R-Tree의 동시성 제어", 석사학위논문, 인하대학교, 2004.