

메인 메모리 상주 GIS 데이터베이스 시스템의 설계

이재호*, 남광우*, 이성호*, 박종현*
*한국전자통신연구원

e-mail : snoopy@etri.re.kr

Design of Memory-Resident GIS Database Systems

Jai-Ho Lee*, Kwang-Woo Nam*, Seong-Ho Lee, Jong-Hyun Park *
*Electronics and Telecommunications Research Institute

요 약

반도체의 가격이 점점 저렴해지고 메모리의 용량이 커짐에 따라 서버 컴퓨터 시스템의 메모리가 대용량화 되고 있다. 이에 따라 많은 양의 공간 데이터를 메모리에 상주 시킬 수 있게 되었다. 또한 LBS 와 같은 고성능을 요구하는 지리 정보를 이용하는 응용들이 나타나고 있다. 본 논문에서는 이와 같은 요구를 만족시키기 위해서 메인 메모리에 상주하는 GIS 데이터베이스 시스템을 다루고자 한다. 메인 메모리 상주 GIS 데이터베이스 시스템은 지리 정보 응용들에서 충분한 기능과 높은 성능을 제공할 것이다.

1. 서론

요즘 LBS, ITS 와 같은 공간 데이터를 이용하는 새로운 응용들이 실시간으로 데이터에 대한 접근과 공간 연산을 필요로 하고 있다. LBS 의 한 예로 점심을 먹기 위해 식당을 찾고 있는 사람에게 찾고자 하는 식당의 위치를 알려 주고 식당으로 가는 길을 알려 주는 서비스를 제공한다면, 식당을 찾는 사람의 현재 위치가 무선 네트워크를 통해 계속 갱신이 되어야 하며 서버는 식당으로 가는 길을 알려 주기 위한 연산을 해야 한다. 이와 같이 빈번한 갱신 작업 수행과 동시에 서비스를 위한 작업들을 수행하기 위해서는 고성능의 데이터베이스에 대한 접근이 필수적이다. 또한 하드웨어 기술의 발달로 점점 메모리의 가격이 낮아지고 서버의 메모리 양이 대용량화 됨에 따라 많은

양의 공간 데이터를 메모리에 상주 시키는 것이 가능해졌다. 따라서 본 논문에서는 고성능의 공간 데이터베이스 시스템을 위해 메인 메모리 기반의 공간 데이터베이스 시스템인 GISDB.MM 을 제안한다.

논문에서는 2 절에서 메인 메모리 DBMS 및 Spatial DBMS 와 관련된 기존 연구를 소개하고 3 장에서는 GISDB.MM 시스템의 전체적인 시스템 구조에 대해서 설명한다. 4 장에서는 공간 데이터 모델에 대해서 5 장에서는 가변길이의 공간 데이터 저장 방법을 기술하고 마지막으로 6 장에서는 결론 및 향후 연구를 언급한다.

2. 관련 연구

메인 메모리 데이터베이스 시스템은 기존의 디스크 기반의 데이터베이스 시스템과는 간략하게 다음과 같

은 차이점이 있다[1]. 디스크는 데이터에 대한 접근 비용이 크고 고정되어 있어 블록단위의 접근 방식을 취하고 있지만 메인 메모리는 아니다. 디스크상의 데이터의 위치가 메인 메모리 보다 매우 중요하다. 왜냐하면 임의의 접근 보다 순차적인 접근이 빠르기 때문이다. 그러나 메인 메모리는 순차적인 접근이 중요하지 않고 프로세서에 의해서 바로 접근이 가능하다. 점점 메모리의 가격이 낮아지고 용량이 커짐에 따라 이러한 특성을 고려한 메인 메모리 기반의 데이터베이스 시스템들이 등장하기 시작하였다. 이에 따라 메인 메모리의 성능을 높이기 위하여 메인 메모리 데이터베이스 시스템에 최적화된 색인, 동시성 제어, 회복 방법들이 많은 연구 결과물로 나오게 되었다.

공간 데이터베이스 시스템에 대해서는 많은 연구가 있었으며 Oracle Spatial, ESRI ArcSDE, DB2 Spatial Extender, Zeus, Informix Spatial Datablade 와 같은 상용화된 여러 데이터베이스 시스템들이 존재한다. 그러나 이러한 데이터베이스 시스템은 대부분 디스크 기반의 DBMS 들이다. 그러나 국내에서 메인 메모리 데이터베이스 시스템에 공간 도메인을 적용하여 시스템을 확장한 Xmas-SX[2]가 있었다. Xmas-SX 는 메인 메모리 저장 구조인 Xmas[3]를 공간 데이터 처리를 위해 확장한 시스템으로 OpenGIS geometry 모델[6]을 따르고 있으며 공간 데이터 타입, 연산자, 색인들을 제공한다.

3. 시스템 구조

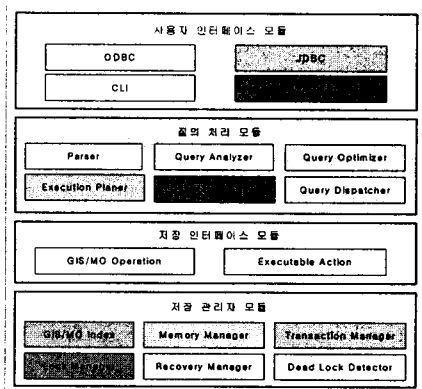


그림 1 GISDB.MM 시스템 구조

그림 1은 GISDB.MM 의 전체적인 구조를 보여 준다. 추후 공간 객체 뿐만 아니라 이동 객체를 다루기 위한 확장을 고려한 구조이다. 또한 GISDB.MM 서버 프로세서는 동시에 여러 트랜잭션을 처리하기 위해 멀티 쓰레드로 동작한다.

각 모듈 별 자세한 내용은 다음과 같다.

3.1 사용자 인터페이스 모듈

GIS 응용 개발자를 위하여 개발언어에 따라 ODBC, JDBC, CLI, 등의 다양한 표준 인터페이스를 제공한다. 또한 OGC 의 OLE/COM 표준[6]을 따르는 데이터제공자를 제공한다. OGC 표준 데이터제공자를 제공함으로써 개방형 구조를 가지게 되며 다양한 응용으로부터 동일한 인터페이스를 통해 DBMS 로의 접근을 제공한다.

3.2 질의 처리 모듈

OpenGIS 에서 정의한 표준 SQL 을 수용한 질의어를 처리하기 위한 질의 처리 모듈은 Query Dispatcher, Parser, Query Analyzer, Execution Planner, Query Optimizer, Catalog Manager 로 구성되어 있다.

Query Dispatcher 는 사용자로부터의 연결 및 질의 요청을 받고 해당 처리모듈로 전달한다. Parser 는 입력된 질의어의 구문 및 의미 분석을 수행하여 파스트리 (parser tree)를 생성한다.

Query Analyzer 는 생성된 파스트리를 분석하고 각 질의 요소에 대한 카탈로그 정보를 확인하며, 정당성 확인을 통해 질의 처리 진행 여부를 결정한다.

Execution Planner 는 각 질의 요소들은 저장관리자에서 실행할 수 있는 형태의 프로시저 행태로 변경되며 이의 실행 순서가 결정된다.

Query Optimizer 는 최적의 실행 경로를 생성하는데 정의된 선택기준(cost-based OR heuristic)에 의해 최적 경로를 선택한다.

Catalog Manager 는 현재 데이터베이스에 생성되어 있는 DB 객체(테이블, 색인, 제약조건 등)에 대한 메타 정보를 관리한다

3.3 저장 인터페이스 모듈

저장 인터페이스 모듈은 GIS/MO operation 모듈과 Executable Action 모듈로 구성되어 있다.

GIS/MO Operation 모듈은 공간 객체에 대한 질의 연산자를 정의하며 질의 처리기에서 사용할 수 있는 인터페이스를 제공한다. 공간 객체에 대한 연산자만으로 구성되어 있으나 추후 이동 객체에 대한 연산자들이 추가 될 것이다. 공간 객체 모델 및 연산자에 대해서는 다음 절에서 자세히 다룬다.

Executable Action 은 실질적으로 저장관리자에서 수행해야 할 일들을 Action 단위로 생성한다. 질의 처리기는 이러한 Action 들의 목록을 생성한다.

3.4 저장 관리자

저장 관리자는 GIS/MO Index, Memory Manager, Transaction Manager, Recovery Manager, Lock Manager, Dead Lock detector 로 구성된다.

GIS/MO index 는 공간 객체에 대한 색인을 제공하고 있으며 추후 시공간 색인이 추가 될 예정이다. 공간 색인으로는 메인 메모리 환경에 맞게 변경한 R*Tree 를 구현하고 비 공간 데이터 대한 색인으로는 T-Tree 와 ECBH 색인을 적용한다.

Memory Manager 는 물리적인 Memory 에 대한 매핑 및 관리를 한다.

Transaction Manager 는 트랜잭션의 처리 및 관리를 담당한다.

Recovery Manager 는 시스템 오류에 대한 데이터베이스의 회복 및 데이터 연속성을 보장한다.

Lock Manager 는 다중 트랜잭션 상에서 테이블간 혹은 레코드간 동시성 제어를 위한 모듈이며 2-Phase Locking protocol 을 적용한다.

마지막으로 Dead Lock Detector 는 시스템의 Dead Lock 을 감지하고 이를 해결하는 역할을 한다.

4. 데이터모델

GISDB.MM 은 공간 정보의 일반적인 표현을 제공하기 위한 OpenGIS geometry model 을 수용하였다. 그

림 2는 OpenGIS Geometry Model 을 따르는 GISDB.MM 의 공간 데이터 타입을 보여준다.

GISDB.MM 에서는 공간 객체들에 필요한 공간 관계 함수들을 모아서 SpatialRelation, 공간 연산자들을 모아서 SpatialOperator, 공간 객체들의 잘 알려진 표현을 다루기 위한 WKS 와 같은 추상 클래스들을 정의하고 있다. SpatialRelation 추상 클래스의 메소드로는 Contains(), Crosses(), Disjoint(), Equals(), Intersects(), Overlaps(), Touches(), Within(), Relate() 등이 있다. SpatialOperator 추상 클래스의 메소드로는 Buffer(), ConvexHull(), Difference(), Distance(), Intersection(), Symdifference(), Union() 등이 있다. 그리고 WKS 추상 클래스의 메소드로는 ExportToWKB(), ExportToWKT(), ImportFromWKB(), ImportFromWKT() 등이 있다.

Point, LineString, Polygon, MultiPoint 등등과 같이 Bold 선으로 둘러싸인 클래스들은 인스턴스로서 생성 가능한 타입들이다. 각각의 공간 객체들에 대한 설명은 OpenGIS Geometry Model[6]에 정의된 것과 거의 동일하며 이미 잘 알려져 있기 때문에 생략한다.

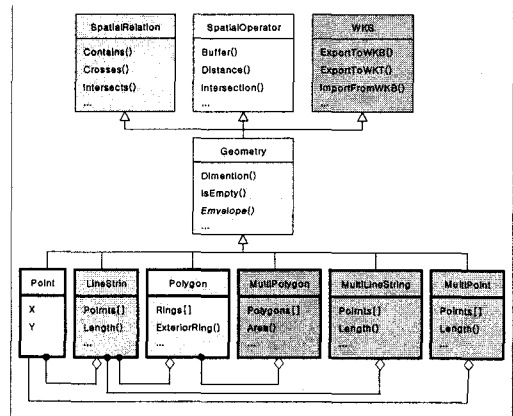


그림 2 GISDB.MM 에서 지원하는 공간 데이터 타입

5. 공간 데이터 관리

일반적으로 공간 데이터는 크기가 일정하지 않다. 따라서 데이터의 크기가 가변적인 공간 데이터를 효과적으로 저장하는 방법이 필요하다. 그림 3은 GISDB.MM 에서 공간 객체를 저장하는 방법을 보여주

고 있다. 데이터베이스는 고정된 크기의 레코드들로 구성되어 있고 레코드들은 고정된 크기의 필드로 구성되어 있다. GISDB.MM에서는 레코드 중 공간 객체를 저장하기 위한 공간 객체 필드가 고정되어 있으므로 공간 객체가 고정된 크기에 저장할 수 없을 경우 BLOB의 형태로 연속된 메모리 공간에 공간 객체를 저장한다.

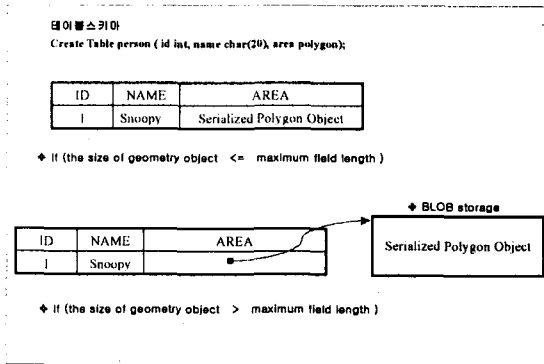


그림 3 공간 객체의 저장 구조

6. 결론 및 향후 연구

본 논문에서는 메인 메모리 기반의 고성능 공간 데이터베이스 시스템을 설계하였다. 시공간 데이터를 다루기 위한 확장을 고려하여 설계되었고 현재 시스템을 구현 중에 있다. 또한 구현 완료 후 다음 단계로서 시공간 데이터를 처리하기 위한 시스템으로 확장할 계획이다. 따라서 대용량의 이동 객체를 다루기 위해 메인 메모리 시스템에 적합한 색인, 동시성 제어, 이동 객체의 저장 방법 등등 다양한 연구들이 필요하다.

참고문헌

- [1] A Hector Garcia-Molina, Kenneth Salem: Main Memory Database Systems: An Overview. TKDE 4(6): 509-516 (1992)
- [2] J.H. Park, K. Kim, S. K. Cha, M.S. Song, S. Lee, and J. Lee, "A High-performance Spatial Storage System Based on Main-Memory Database Architecture", In Proceedings of DEXA Conference, 1999
- [3] J. H. Park, Y. Sik Kwon, K. Kim, S. Lee, B. D. Park, and

- S. K. Cha, "Xmas: An Extensible Main-memory Storage System for High-Performance Applications", In Proceedings of ACM SIGMOD Conference, 1998.
- [4] H. V. Jagadish, D. Lieuwen, R. Rastogi, and A. Silberschartz, "Daili: A High Performance Main Memory Storage Manager", In Proceedings of VLDB Conference, 1998
- [5] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD Conference 1990: 322-331
- [6] OpenGIS Implementation Specifications : OpenGIS Simple Features Specification For OLE/COM Revision 1.1, OpenGIS Consortium, Inc. 1999
- [7] OpenGIS Implementation Specifications : OpenGIS Simple Features Specification for SQL Revision 1.1, OpenGIS Consortium, Inc. 1999