

GEO/Millennium: 클라이언트-서버 공간 데이터베이스 시스템

이환재, 안준순*, 강동세*, 이경모*, 정보홍*,

박동선*, 배해영*

*인하대학교 전자계산공학과

GEO/Millennium: Spatial Database System in Client-Server Environment

Hwan-Jae Lee, Jun-Soon Ahn*, Dong-Jae Kang*, Kyung-Mo Lee*, Bo-Heung Jung*,

Dong-Seon Park*, Hae-Young Bae*

*Dept. of Computer Science & Engineering, Inha University

요 약

대용량의 공간 및 비공간 데이터의 관리와 다수 사용자에 의한 데이터 공유를 지원하기 위해서는 클라이언트-서버 구조의 공간 데이터베이스 시스템을 사용하는 것이 효과적이다.

본 논문에서는 클라이언트-서버 구조의 공간 데이터베이스 시스템인 GEO/Millennium 시스템을 소개한다. GEO/Millennium은 클라이언트 캐칭을 수용한 질의 전송 방식을 사용하며, 공간 및 비공간 데이터를 통합하여 저장, 관리하는 통합구조의 시스템이다. 제한하는 시스템은 데이터베이스의 변경사항을 멀티캐스팅으로 클라이언트들에게 통지하여 데이터의 일관성을 유지하며, 클라이언트의 각 모드를 컴포넌트로 구성하여 어플리케이션 개발의 용이성과 재사용성을 높인다.

GEO/Millennium은 공간 객체의 저장과 연산을 저장관리자와 질의 처리기에서 지원해 줌으로써 공간연산을 효율적으로 수행할 수 있으며, 클라이언트-서버간의 통신부하를 줄여 클라이언트의 질의에 대해 보다 빠른 응답시간을 갖는다.

1. 서론

국가적인 NGIS사업의 추진 등 최근 국내 공간 데이터베이스의 구축 규모의 증대와 인터넷 및 네트워크의 확산으로 공간 데이터베이스를 필요로 하는 분야가 점차 다양해지고 있다. 이렇게 구축된 데이터베이스를 보다 많은 사용자와 공유하기 위해서는 클라이언트-서버 구조의 공간 데이터베이스 시스템을 사용하는 것이 효과적이다[1].

본 논문에서는 인하대학교 데이터베이스 연구실에서 개발중인 공간 데이터베이스 시스템인 GEO/Millennium 시스템을 소개한다. 제한하는 시스템은 공간 및 비공간 데이터에 대한 저장 및 연산을 모두 하나의 시스템에서 수행하는 통합구조로 설계되었다. 질의처리 방식으로는 질의전송기법을 사용하였으며, 간단한 질의는 클라이언트에서 처리할 수 있도록 클라이언트 캐칭을 사용한다. 캐칭을 사용할 경우 클라이언트가 캐시 데이터를 사용하기 전에 유효성을 검사하는 단계가 필요한데, 제한된 시스템은 이런 단계의 부하를 줄이기 위해 데이터베이스가 변경되면 서버가 클라이언트에게 변경사항을 통지하도록 하여 캐시 데이터의 일관성을 유지시킨다. 그리고, 전송된 객체의 리스트를 서버에서 관리하여 이후에 질의의 결과로 이미 전송된 객체가 다시 전송되는 것을 방지한다.

제한하는 시스템은 통합구조로 설계되어 데이터의 효율적인 관리와 질의 최적화가 가능하며, 트랜잭션의 관리와 회복이 용이하다. 또한, 클라이언트와 서버간의 통신부하를 줄여 클라이언트의 질의에 대해 보다 빠른 응답시간을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 공간 데이터베이스 시스템의 구조와 클라이언트-서버간의 질의 처리 방법들을 설명한다. 3장에서는 GEO/Millennium 시스템의 실제 시 성능향상을 위해 고려한 점들을 기술하고, 4장에서는 GEO/Millennium 시스템의 구조에 대한 개략적인 설명을 한다. 마지막으로 5장에서 결론 및 향후

연구과제를 제시한다.

2. 관련 연구

2.1 공간 데이터베이스 시스템의 구조

공간 데이터베이스의 구조는 공간특성을 데이터베이스와 통합한 방법에 따라 크게 3가지로 나뉜다[2].

계층구조(Layered Architecture)의 공간 데이터베이스 시스템은 기존의 상용 DBMS의 대용량 객체(Large Object) 타입에 공간 데이터를 저장하고, DBMS위에 추가된 계층에서 공간연산을 처리하는 시스템이며, GEOVIEW, SIRO-DBMS 등이 대표적인 예이다 [2]. 그러나, 이러한 시스템에서는 공간 데이터가 단순히 바이트 스트림에 저장되어 의미를 가지지 못하고, 공간데이터에 대한 연산이 공간 데이터를 구현하는 최상위의 계층에서 이루어지기 때문에 효율성이 떨어지게 된다.

이중구조(Dual Architecture)의 공간 데이터베이스 시스템은 비공간 데이터는 기존의 DBMS에 저장하며 공간 데이터는 파일 기반의 저장 시스템에 저장한다. 그리고, 이 두 시스템을 통합 조정하는 상위계층이 질의를 둘로 나누어 처리한다. 대표적인 구현 예로는 ARC/INFO, SICAD등이 있다[2]. 그러나, 이러한 시스템들은 하나의 객체를 둘로 나누어 저장하므로 서로 연결할 링크가 필요하고, 공간객체에 대한 질의 역시 서로 다른 두 시스템에서 처리하므로 나누어진 두 질의의 비용에 대한 비교를 할 수가 없어 질의 최적화를 수행할 수 없다.

통합구조(Integrated Architecture)의 공간 데이터베이스 시스템은 주로 확장 가능한 DBMS를 이용해서 구현되는데, 질의 처리기가 공간객체를 위한 ADT(Abstract Data Type)를 제공하여 공간 데이터와 비공간 데이터의 저장 및 연산에 있어서 일관적인 통합처리가 가능하다. 대표적인 구현 예로는 Probe, DASDBS 등이 있다[2]. 이 구조는 공간 및 비공간 데이터간의 링크가 필요 없고,

* 본 연구는 정보통신부의 대학 S/W 연구센터 지원사업의 연구결과임.

공간 데이터의 특성에 맞춘 인덱스와 연산 알고리즘의 사용이 가능하여 효율적인 공간연산을 수행할 수 있다.

2.2 클라이언트-서버간의 질의 처리방식

일반적인 클라이언트-서버 시스템은 질의처리 방식에 따라 크게 2가지로 나뉜다[4].

질의 전송(Query Shipping) 방식은 클라이언트가 질의를 서버로 보내면 서버가 질의를 수행한 후 결과를 클라이언트로 전송하는 방식이다. 이 방식은 서버가 클라이언트에게 질의에 대한 결과만 전송하므로 서버와 클라이언트간의 통신부하가 적은 장점이 있는 반면, 모든 질의가 서버로 집중되기 때문에 서버 측에 병목현상이 발생하는 단점이 있다.

데이터 전송(Data Shipping) 방식은 클라이언트가 자체적으로 질의를 처리하기 위해 질의처리 시에 필요한 데이터를 서버에 요청하여 캐칭하는 방식이며, 서버에 요청하는 데이터의 단위는 주로 파일이나 페이지이다. 이 방식은 전송의 단위가 커서 질의에 필요하지 않은 데이터도 보내지게 되어 서버와 클라이언트간의 통신부하가 증가하는 단점이 있다.

3. GEO/Millennium 시스템의 설계

3.1 통합구조의 공간 데이터베이스 시스템

계층구조나 이중구조의 공간 데이터베이스의 경우, 공간 및 비공간 데이터의 저장과 연산이 동일하게 처리되지 못하여 질의 처리의 효율성이 떨어지며, 질의 비용의 상대적 비교도 불가능하여 질의 최적화를 할 수 없다. 따라서, GEO/Millennium은 공간 데이터의 효율적인 처리를 위해 공간 데이터와 비공간 데이터가 동일한 시스템 내에서 일관적으로 저장 및 연산되는 통합구조로 설계한다.

통합구조로 설계된 본 시스템은 질의 처리를 효율적으로 할 수 있을 뿐만 아니라, 질의가 둘로 나뉘어지지 않아 계층구조나 이중구조의 공간 데이터베이스 시스템보다 트랜잭션의 관리 및 동시성 제어, 그리고 회복에도 유리한 장점을 가진다.

3.2 클라이언트-서버간의 통신비용 감축

GEO/Millennium 시스템이 클라이언트-서버간의 통신비용을 감축시키기 위해 다음의 4가지를 고려하였다.

첫째, 클라이언트-서버간의 질의 처리 방식으로는 데이터 전송방식보다 통신부하가 적은 질의 전송(Query Shipping) 방식을 채택하였다. 질의는 헤더정보와 명령어, 그리고 필요한 데이터에 대한 파라미터를 가지는 패킷으로 구성되어 서버로 전송함으로써 클라이언트가 서버의 API함수를 호출하는 형태를 가진다. 서버는 이 질의를 수행한 후 결과 테이블과 객체들의 고유한 ID를 함께 클라이언트에게 전송하며, 클라이언트는 전송된 데이터를 저장하고 객체의 ID로 인덱스를 구성하여 이후의 질의 시에 사용한다.

둘째, 질의 전송 방식으로 인해 서버로 부하가 집중되는 것을 방지하기 위하여 클라이언트 캐칭을 이용한다. 질의 전송 방식에서 클라이언트 캐칭을 사용할 경우, 결과가 중복되는 질의들을 보내면 이미 전송된 데이터가 다시 전송되어 불필요한 통신과 저장공간의 중복을 유발할 수가 있다. 이를 방지하기 위하여 GEO/Millennium 시스템은 전송된 객체에 관한 정보를 서버가 관리하도록 하여, 클라이언트에게 전송되지 않은 객체는 데이터와 ID를 함께 전송하고 이미 전송된 데이터에 대해서는 객체의 ID만 전송한다. 그리하여, 클라이언트는 전송된 객체들의 데이터가 아닌 ID로 질의 결과를 구성한다.

셋째, 클라이언트가 캐시 데이터를 이용하여 질의처리를 할 경우, 매 질의마다 캐칭되어 있는 데이터가 현재 서버에 있는 데이터와 일치하는지에 대해 검사하는 단계가 필요하다. GEO/Millennium 시스템은 이 단계를 제거하기 위해서 서버의 데이터베이스에 갱신이 일어나면 그 데이터베이스에 접근중인 모든 클라이언트에게 변경된 사항에 관한 통지 메시지를 보내준다. 메시지를 받은 클라이언트는 변경된 데이터를 서버로부터 다시 전송 받을 수 있는 질의를 보내고 결과를 반영함으로써 캐시 데이터와 서버의 데이터베이스의 일관성을 항상 유지하도록 한다.

넷째, 스타일 테이블(Style Table)의 개념을 도입한다. 스타일이란 공간객체가 GUI(Graphic User Interface)를 위해 가지는 속성정보로써, 선의 색깔이나 굵기, 면의 색이나 패턴 등의 정보이다. 공간 데이터는 레이어(Layer) 단위로 나뉘어 관리되는데, 한 레이어는 한정된 몇 가지 종류의 객체들로만 구성되며, 같은 종류의 객체들은 같은 스타일 정보를 가진다. GEO/Millennium 시스템은 각 레이어별로 공간 객체들의 스타일 정보의 종류를 분류하여 그 레이어에 속한 스타일 테이블에 저장하고 그 종류마다 스타일 ID를 부여하며, 각각의 공간객체에게 이 스타일 ID를 외래키로 가지게 한다. 그리하여, 공간객체가 저장되는 공간을 절약하고, 또한 클라이언트-서버간의 통신비용도 줄일 수 있도록 한다.

3.3 효율적인 질의결과 전송

디스크에 대한 접근과 메모리 연산의 횟수는 시스템의 성능에 큰 영향을 미친다. GEO/Millennium 시스템은 결과 테이블에 대한 디스크 접근과 메모리 연산을 줄이기 위해 질의의 결과로 나온 레코드를 메모리에 패킷형태로 저장한다. 그리고, 모든 결과 레코드를 메모리에 저장하면 메모리를 많이 차지하여 서버의 성능을 저하시키는 요인이 되기 때문에, 패킷의 크기가 최대크기를 넘어서면 해당 패킷을 전송한 후 그 메모리 공간에서 패킷을 계속 구성한다. 질의처리가 완료되어 결과 테이블이 다 만들어질 때까지 기다리지 않고, 패킷을 구성하는 대로 클라이언트에게 보내는 것은 사용자에게 좀더 빨리 결과를 보내주어 응답속도를 높이는 효과도 기대할 수 있다.

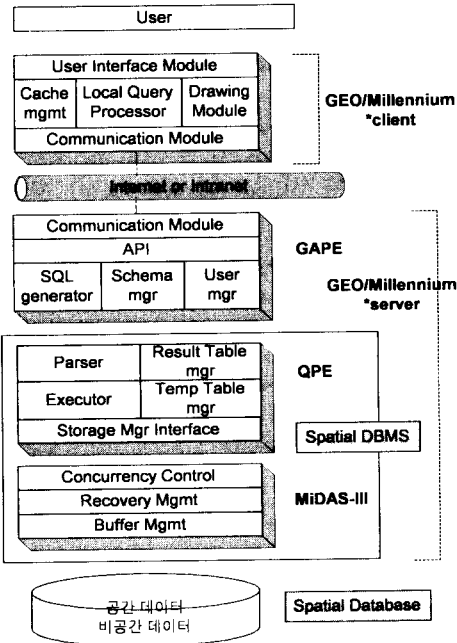
3.4 클라이언트 모듈의 컴포넌트화

공간 데이터베이스 시스템을 다양한 분야에서 활용할 수 있게 하기 위해서는 공간 데이터베이스 시스템의 클라이언트로서 가져야 할 기본적인 GUI 및 통신기능 뿐만 아니라 응용 분야에 필요한 기능도 추가적으로 갖는 클라이언트를 쉽게 개발할 수 있어야 한다. GEO/Millennium 시스템은 클라이언트 어플리케이션 개발의 용이성과 재사용성을 높이기 위하여 기본적인 기능을 갖는 모듈을 독립적인 DLL(Dynamic Linking Library) 컴포넌트로 구현하여 어플리케이션 개발환경인 ADE(Application Development Environment)를 제공한다. 프로그래머는 필요한 컴포넌트를 조합한 후 제공하는 인터페이스를 이용하여 쉽게 프로그래밍 할 수 있으며, 향후 클라이언트의 기능확장도 용이하게 할 수 있다.

4. GEO/Millennium 시스템의 구현

4.1 GEO/Millennium 시스템의 구성

GEO/Millennium 시스템은 [그림 1]과 같이 저장관리자, QPE, GAPE, 그리고 클라이언트로 구성되어 있으며, 각 클라이언트들의 개별적 관리를 용이하게 하기 위해 서버 프로세스와 클라이언트의 대응관계를 1:1이 되도록 구현하였다.

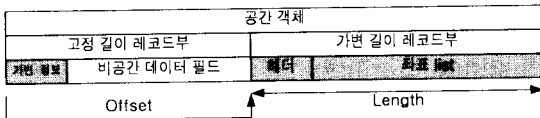


[그림 1] GEO/Millennium 시스템의 구조

4.1.1 저장관리자

GEO/Millennium 시스템은 저장관리자로 한국전자통신연구원(ETRI)에서 만든 인덱스 기반의 객체관계형 DBMS인 MIDAS-III를 사용한다[3].

저장관리자는 하나의 공간객체에 대해 [그림 2]와 같이 공간 데이터와 비공간 데이터를 하나의 레코드로 구성하여 저장한다. 공간 객체의 저장구조는 비공간 속성에 대한 접근을 쉽게 하기 위해 고정 길이 레코드 부와 가변 길이 레코드 부로 구분한다. 고정 길이의 레코드 부는 고정길이 레코드 부분의 길이, 가변길이 레코드 부분의 길이, MBR, 스타일 ID의 4개의 기본적인 공간 속성과 추가적인 비공간 속성들을 가지고, 가변 길이 레코드 부는 공간객체의 타입과 좌표의 개수, 그리고 실제 좌표데이터를 저장하기 위한 배정도형을 요소로 갖는 리스트를 가진다.



[그림2] 공간객체의 저장구조

4.1.2 QPE(Query Processing Engine)

QPE는 SQL 질의를 파싱한 후 저장관리자가 지원하는 인터페이스를 이용하여 질의를 수행하고 결과를 패킷으로 만들어 클라이언트에게 전송하는 역할을 한다. QPE는 SQL2를 기반으로 하며 공간연산을 할 수 있도록 공간 객체 타입과 공간 연산자가 확장된 질의어를 사용하는데, G_POINT, G_LINE, G_RECT, G_CIRCLE, G_POLYLINE, G_POLYGON등의 공간 객체 타입과 INTERSECT, DISJOINT, CONTAIN, EQUAL, TOUCH, CROSS, OVERLAP 등

의 공간 연산을 지원한다.

4.1.3 GAPE(Geo-spatial API Processing Engine)

GAPE는 효율적인 전송과 클라이언트 캐시 데이터의 일관성 유지를 위해 클라이언트의 주소, 사용자 ID, 접근중인 데이터베이스, 전송된 객체의 리스트 등을 관리한다. 그리고, 사용자로부터 받은 패킷 형태의 질의를 공간 데이터의 스키마에 맞는 SQL 질의문으로 변환하여 QPE에 전달하는 역할을 한다.

4.1.4 클라이언트

GEO/Millennium의 클라이언트는 통신모듈, 캐시 관리 모듈, 드로잉 관리 모듈, 그리고 사용자 인터페이스 모듈로 구성되며, 다양한 공간 데이터베이스 응용 프로그램의 개발이 용이하도록 각각 모듈이 독립된 DLL로 구성되어있다.

통신모듈은 서버와의 통신을 담당하는 모듈로서, 전송 받은 결과를 저장하여 캐시 관리 모듈에 전달하며, 서버로부터의 여러 가지 통지메시지를 받는 역할을 수행한다. 캐시 관리 모듈은 전송 받은 데이터를 관리하여 간단한 질의는 클라이언트가 자체적으로 수행할 수 있도록 한다. 드로잉 관리 모듈은 공간연산을 위한 편리한 인터페이스를 제공하며, 질의 결과를 사용자에게 쉽게 전달할 수 있는 시각적 수단을 제공한다. 사용자 인터페이스 모듈은 프로그래머가 위의 세 모듈을 간단하게 쓸 수 있도록 인터페이스를 제공한다.

5.결론 및 향후 연구방향

본 논문에서는 클라이언트-서버 구조의 공간 데이터베이스 시스템인 GEO/Millennium 시스템을 제안하였다. GEO/Millennium은 공간 및 비공간 데이터를 통합하여 저장, 관리하는 통합구조의 시스템이며, 클라이언트 캐싱을 수용한 질의 전송 방식을 사용했다. 제안한 시스템은 데이터베이스의 변경사항을 멀티캐스팅으로 클라이언트들에게 통지하여 데이터의 일관성을 유지하며, 클라이언트의 각 모듈을 컴포넌트로 구성하여 어플리케이션 개발의 용이성과 재사용성을 높였다.

제안한 시스템은 통합구조로 설계되어 공간 및 비공간 데이터의 효율적인 관리와 공간 연산과 비공간 연산의 통합된 질의 최적화가 가능하며, 클라이언트 캐시의 사용과 멀티캐스팅을 통한 캐시 효율성의 제공으로 클라이언트와 서버간의 통신부하를 최소화하였다.

향후 연구과제로는 여러 서버에 분산된 공간 데이터에 대한 효율적인 처리를 위해 분산환경에서 서버 및 클라이언트의 역할분담에 관한 연구가 필요하다.

참고문헌

[1] Dr. Manfred von Seggern, "The Enterprise GIS: A Client/Server Approach Using Distributed Relational Databases To Create Multi Location/Multi Application GIS," EGIS, vol. 1, pp.652-659, 1994.
 [2] Güting, R., "An Introduction to Spatial Database Systems," VLDB Journal, Vol.3, pp.357-399, 1994.
 [3] Kim, P.C., Choi H.I., Lee Y.,J., Lee S.H., and Kim M.J., "MIDAS: Design philosophy and Internals," IPCC, pp.132-139, 1992.
 [4] 장규원, "클라이언트-서버 환경을 지원하는 GIS에서의 역할분담 스케줄러," 인하대학교 전자계산공학과 석사논문, 1998.