

Mapbase 컴포넌트에서 공간 데이터를 위한 버퍼 관리 전략

조대수⁰, 김광수, 이종훈
한국전자통신연구원, 공간·영상정보기술센터, GIS 팀
e-mail: {junest, enoch, jong}@etri.re.kr

A Buffer Management Strategy for Spatial Data in Mapbase Components

Dae-Soo Cho⁰, Kwang-Soo Kim, Jong-Hun Lee
GIS Team, Spatial & Visual Information Technology Center, ETRI

요 약

서로 다른 GIS 서버의 공간 데이터를 동일한 방법으로 접근하기 위한 개방형 GIS 에 관심이 커지고 있다. 서로 다른 서버간의 상호 운용성을 효과적으로 지원하기 위해서는 데이터제공자를 통한 공간 데이터 모델의 변환이 필수적이므로, 방대한 공간 데이터의 변환과 전송으로 인한 사용자 응답 시간의 지연 문제가 우선 해결되어야 한다.

이 논문에서는 OGIS 데이터소비자에서 효과적인 버퍼 관리를 통해 사용자 응답 시간의 지연 문제를 해결한다. 이 논문에서 제시하는 버퍼 관리 전략은 전체 레이어에 대해서 통합적으로 메모리를 관리하고 각 레이어별로 공간 인덱스를 생성·관리함으로써 데이터 접근 속도를 개선하였다. 제안한 버퍼 관리 전략은 OGIS 데이터소비자인 Mapbase 컴포넌트를 통해 구현함으로써, 데이터 접근 속도의 개선을 입증하였다.

1. 서론

개방형 구조의 클라이언트/서버 GIS 에 대한 수요가 증가함에 따라서 표준화에 대한 중요성이 부각되고 있다. OGC(Open GIS Consortium)의 심플 피처 구현 명세는 개방형 GIS 구조에 대한 산업체(de facto) 표준을 제시하고 있다. 그리고 OGC 의 산업체 표준은 현재 ISO 19125 를 통해 국제표준으로 제정되기 위한 절차가 진행되고 있다.

GIS 서버로부터 공간 데이터를 검색, 획득, 변경할 수 있음을 의미한다. 즉, 사용자가 하나의 클라이언트를 통해서 서로 다른 GIS 서버의 데이터에 접근할 수 있음을 의미한다. 이러한 개방형 GIS 구조의 가장 큰 특징은 상호 운용성을 통해 각 기관별로 구축되어 있는 공간·비공간 데이터를 쉽게 공유할 수 있다는 점이다.

그림 1은 OGC 에서 제안하는 OLE/COM 기반의 개방형 GIS 구조를 보이고 있다. OGIS 데이터소비자(OGIS Data Consumer)는 각 GIS 서버의 고유한 인터페이스를 알 필요 없이, 표준 인터페이스만을 이용해서 서로 다른 GIS 서버에 접속하여, 데이터에 접근할 수 있음을 보이고 있다. 표준 인터페이스는 마이크로소프트(Microsoft®)의 OLE DB 인터페이스를 기반으로, 기하 데이터 접근을 위해 OGC 에서 추가로 인터페이스를 확장한 것을 의미한다. 각각의 GIS 서버 자체는

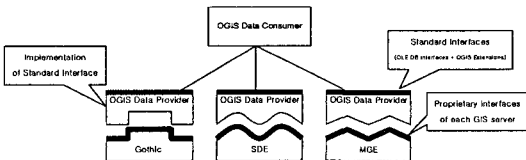


그림 1 OGC 기반의 개방형 GIS 구조

개방형 GIS 구조는 동일한 방법으로 서로 다른

표준 인터페이스를 제공하지 않는다. 따라서 각 서버별로 표준 인터페이스를 구현한 컴포넌트인 OGIS 데이터제공자(OGIS Data Provider)를 제공해야 한다.

OGIS 데이터제공자는 해당 GIS 서버가 사용하는 데이터 모델에 관계없이, 관계형 모델의 테이블 형태(OLE DB에서는 로셋(Rowset)이라고 함)로 OGIS 데이터소비자에게 데이터를 제공해준다. 즉, OGIS 데이터제공자는 내부적으로 각 GIS 서버가 지원하는 자체 데이터 모델을 표준 데이터 모델로 변환하는 과정을 거친다. 따라서 데이터에 접근하는 속도가 저하되는 문제를 구조적으로 포함하게 된다. 대신에, 이러한 데이터 변환 비용을 통해 서로 다른 GIS 서버간의 상호 운용성이 달성되는 것이다.

이 논문은 개방형 GIS 구조에서 공간 데이터의 버퍼 관리(Buffer Management)를 다룬다. 버퍼 관리는 느린 데이터 접근 속도의 문제를 간접적으로 해결할 수 있는 방법이다. 개방형 GIS 구조에서 공간 데이터의 버퍼 관리가 어려운 이유는 첫째, 공간 데이터의 양이 매우 크기 때문이며, 둘째, GIS 서버에서 관리되던 공간 데이터에 대한 메타 정보 및 공간 인덱스 등이 OGIS 데이터제공자를 거치면서 모두 사라지기 때문이다.

이 논문의 구성은 다음과 같다. 2장에서는 Mapbase 컴포넌트의 구조와 문제점을 살펴본다. 3장에서는 버퍼 관리 전략을 설명하고, 4장에서 결론을 맺는다.

2. Mapbase 컴포넌트

Mapbase 컴포넌트는 한국전자통신연구원에서 개발한 OGIS 데이터소비자로서, 크게 데이터제공자와의 연결을 담당하는 컴포넌트와 화면출력을 하는 컴포넌트로 구성된다. 그림 2는 Mapbase 컴포넌트의 개략적인 구성과 역할을 보이고 있다. 연결을 담당하는 컴포넌트는 여러 개의 레이어 객체(Layer)들로 구성되는 레이어 컬렉션 객체(Layers)를 가지고 있다. 레이어 컬렉션 객체 내의 레이어 객체는 출력 우선순위(display priority)를 갖는다. 화면 출력을 담당하는 컴포넌트에는 지도출력(DapDraw) 객체와 출력될 피쳐들의 스타일을 설정하기 위한 객체들이 포함되어 있다.

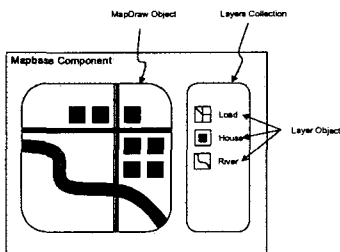


그림 2 Mapbase 컴포넌트의 구성

하나의 레이어 객체는 특정 데이터제공자가 제공하는 하나의 피쳐 테이블(Feature Table)과의 연결정보와 출력 여부에 대한 정보를 저장하고 있다. 피쳐 테이블은 OGIS 데이터제공자를 통해 접근 가능한 테이블로써, 기하 데이터를 포함하고 있다. 연결정보에는 데이터제공자 이름, 피쳐 테이블 이름, 제약 조건 등이 포함되어 있다. 제약조건은 하나의 레이어 객체가 특정 피쳐 테이블의 전체 데이터가 아닌, 이 조건을 만족하는 일부 데이터와 연결될 수 있도록 한다.

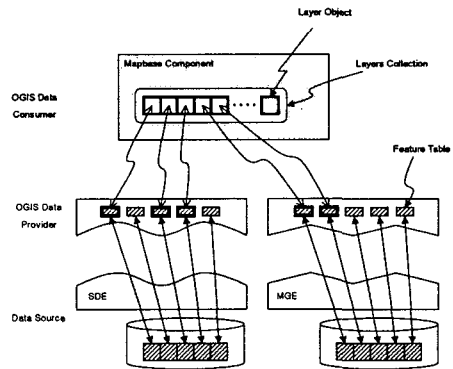


그림 3 Mapbase 컴포넌트의 레이어 연결 예

Mapbase 컴포넌트는 개방형 GIS 구조를 따르기 때문에, 그림 3과 같이 OGIS 데이터제공자를 통해서 서로 다른 GIS 서버와 동시에 연결하여 데이터에 접근할 수 있는 장점을 갖는다. 즉, Mapbase 컴포넌트는 GIS 서버의 종류에 관계없이 동일한 방법으로 데이터에 접근할 수 있으므로, OGIS 데이터제공자를 제공하는 모든 GIS 서버의 데이터를 공유할 수 있다.

반면에, Mapbase 컴포넌트와 같은 개방형 GIS 구조에서는 데이터에 접근하는 속도가 느려지는 단점을 갖는다. 그림 3에서 보는 것과 같이 Mapbase 컴포넌트는 서로 다른 GIS 서버간의 상호 운용성을 달성하기 위해서 각 서버별로 작성된 OGIS 데이터제공자를 통해 데이터에 접근해야 한다. 즉, 각 OGIS 데이터제공자에서는 해당 GIS 서버의 데이터에 접근해서 표준 인터페이스에 맞는 데이터로 변환하고, Mapbase 컴포넌트는 표준 인터페이스에 따라 OGIS 데이터제공자에 의해 변환된 데이터에 접근한다. 따라서, Mapbase 컴포넌트가 특정 GIS 서버의 데이터에 접근하기 위해서는 내부적으로 두 번의 연결과정과 두 번의 데이터 접근과정이 필요하므로, 데이터 접근 속도가 느려지게 된다.

OGIS 데이터소비자와 같은 GIS 응용프로그램에서는 다음과 같이 두 가지 유형의 데이터 접근이 발생한다. 첫째, 공간 데이터의 화면 출력을 위해서, 출력 설정된 모든 레이어의 공간 데이터에 접근한다. 둘째,

공간·비공간 질의의 결과를 얻기 위해서, 해당 질의의 결과 데이터에 접근한다. 화면 출력을 위한 공간 데이터 요청은 공간 연산 등의 질의 요청에 비해 매우 빈번히 발생될 뿐 아니라, 전송되는 데이터의 양도 상대적으로 매우 많다. 따라서 데이터 접근 속도가 느린 개방형 GIS 구조에서는 화면 출력을 위한 공간 데이터의 요청에 대한 응답 시간의 지연이 발생하게 된다.

이 논문에서는 공간 데이터의 화면 출력 시의 사용자 응답 시간을 줄이기 위한 공간 데이터의 버퍼 관리를 다룬다. 비공간 데이터의 버퍼 관리 및 공간·비공간 질의를 위한 버퍼 관리는 이 논문의 대상이 아니다. 먼저, Mapbase 컴포넌트에서 공간 데이터의 화면 출력과정을 살펴보면 다음과 같다.

1. 사용자의 요청에 따라 출력될 화면에 변경이 발생하면 화면 출력을 시작한다.
2. Mapbase 컴포넌트의 지도출력 객체는 출력 우선순위에 따라 다음으로 출력되어야 할 레이어 객체를 선택한다.
3. 선택된 레이어 객체의 연결 정보를 이용하여 해당 GIS 서버의 OGIS 데이터제공자로부터 공간 데이터를 가져온다.
4. 가져온 공간 데이터는 화면 출력 후 메모리에서 제거된다.
5. 레이어 컬렉션에 출력할 레이어가 남은 경우에 2.4 을 반복한다.

출력될 화면이 변경될 때마다 OGIS 데이터제공자를 통해 공간 데이터를 재요청 해야 하는 이유는 Mapbase 컴포넌트가 가져온 공간 데이터에 대한 메모리 관리를 하지 않고, 화면 출력 후에 모두 메모리에서 제거하기 때문이다. Mapbase 컴포넌트에서 화면 출력을 위한 공간 데이터의 요청은 사용자에게 의해 출력될 화면의 내용이 변경될 때 발생하며, 다음과 같이 크게 세가지로 분류된다. 그런데, OGIS 데이터제공자로부터 가져온 공간 데이터를 화면 출력 후 제거하지 않고, Mapbase 컴포넌트가 공간 데이터 버퍼에 관리한다면, 두 번째와 세 번째 경우에는 OGIS 데이터제공자를 통해 공간 데이터를 가져올 필요가 없어진다.

1. 새로운 레이어를 화면 출력하기 위해 특정 OGIS 데이터제공자가 지원하는 피쳐 테이블과의 연결 설정을 위해 레이어 객체를 생성할 때
2. 화면 조작(이동, 확대, 축소 등)에 따라 현재 출력 중인 화면의 영역이 변경될 때
3. 연결 설정된 레이어 객체에 대한 출력 여부가 변경되거나 출력 우선순위가 변경될 때

3. 버퍼관리

이 장에서는 Mapbase 컴포넌트에서 공간 데이터의

버퍼 관리를 위한 구조를 살펴본다. 그림 4는 버퍼 관리를 위해 각 레이어 객체별로 버퍼관리자(BM: Buffer Manger) 객체와 디스크관리자(DM: Disk Manger) 객체를 추가한 것을 보이고 있다. 각 레이어는 읽어들인 공간 데이터를 버퍼관리자를 통해 버퍼에 저장한다. 버퍼관리자는 자체적으로 메모리버퍼와 디스크관리자를 가지므로 메모리버퍼의 크기만큼 공간 데이터를 저장하고, 메모리버퍼가 부족한 경우에 디스크관리자를 통해 디스크버퍼에 저장한다.

하나의 레이어 객체가 새로 생성되면, 레이어 객체의 연결 정보를 기반으로 공간 데이터를 읽어온다. 이때, 읽어들인 공간 데이터를 화면 출력 후 메모리에서 제거하지 않고, 버퍼관리자를 통해 버퍼에 저장하게 된다. 따라서, 다음 번 공간 데이터 요청 시에는 OGIS 데이터제공자와 연결하지 않고, 자체 버퍼로부터 공간 데이터를 가져오게 된다.

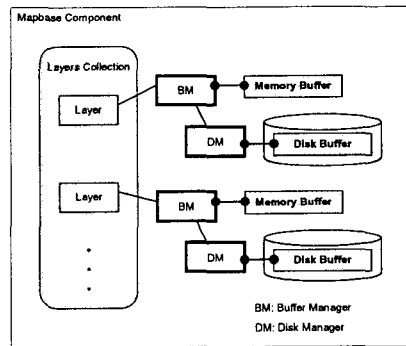


그림 4 버퍼 관리를 위한 간단한 구조

화면 출력을 위한 공간 데이터의 요청 횟수를 줄이는 것은, 버퍼관리자와 디스크관리자를 Mapbase 컴포넌트에 추가함으로써 쉽게 달성될 수 있다. 즉, Mapbase 컴포넌트는 레이어 객체를 새로 생성할 때만 OGIS 데이터제공자에게 공간 데이터를 요청한다.

그림 4와 같은 버퍼 관리는 다음과 같은 문제를 갖는다. 첫째, 각 레이어의 버퍼관리자가 관리하는 메모리버퍼의 크기를 설정하기가 매우 어렵다. 만약, 가변 길이의 메모리버퍼를 사용한다면, 메모리 버퍼의 전체 크기를 관리할 수 없다. 따라서 메모리버퍼를 할당 받지 못하는 레이어 객체는 모든 공간 데이터를 디스크 버퍼에 저장해야 하며, 메모리 부족으로 한번에 하나의 공간 객체만을 디스크로부터 읽어와야 하므로 성능이 매우 저하된다. 반면에 고정 길이의 메모리버퍼를 사용한다면, 생성 가능한 레이어 객체의 수가 고정되며, 레이어 객체가 최대 생성되기 전까지는 메모리의 낭비가 발생하게 된다.

따라서 각 레이어 객체별로 할당된 모든 메모리버퍼를 전체적으로 관리하게 위한 모듈이 필요하다. 이

논문에서는 그림 5와 같이 메모리관리자(MM: Memory Manger) 객체를 통해 전체 메모리버퍼를 관리하게 된다. 메모리관리자는 버퍼관리자가 메모리버퍼를 요청하면 메모리를 할당하는 역할과 메모리가 부족한 경우에 특정 버퍼관리자의 메모리버퍼를 회수하는 역할을 담당한다.

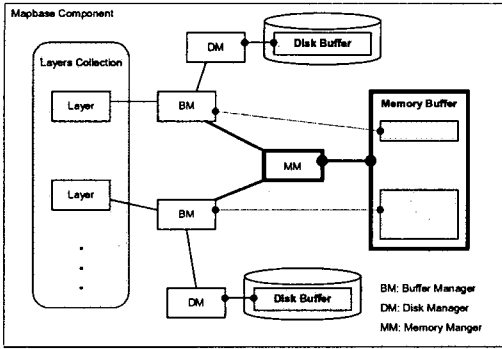


그림 5 효율적인 메모리 관리를 위한 메모리관리자

사용자의 요청에 따라 출력될 화면에 변경이 발생하면, 새로 추가된 레이어 객체는 해당 OGIS 데이터 제공자를 통해서, 기존의 레이어 객체는 자신의 버퍼관리자를 통해서 공간 데이터를 얻어오게 된다. 이때, 버퍼관리자를 통해서 공간 데이터를 얻어오는 경우에 추가의 성능 개선이 가능하다.

현재 메모리 및 디스크버퍼에 저장된 공간 객체에 대해서는 공간 인덱스가 없기 때문에, 사용자가 화면 출력을 원하는 영역에 상관없이 모든 데이터를 읽어와야 한다. 따라서, 메모리버퍼는 단지 디스크버퍼에 있는 공간 데이터를 읽어오기 위한 수단으로만 사용된다. 처음 레이어 객체가 생성되어 OGIS 데이터제공자로부터 공간 데이터를 읽어올 때, 공간 인덱스를 생성하고 공간 인덱스의 말단 노드별로 공간 데이터를 묶어서 관리한다면, 공간 인덱스를 이용한 여과 과정을 거쳐 일부 공간 데이터만 읽어오면 된다. 또한, 사용자에 의한 출력 화면의 변경은 공간 지역성을 가지고 있으므로, 디스크버퍼에 대한 접근 횟수가 줄어들어 성능 향상을 가져온다.

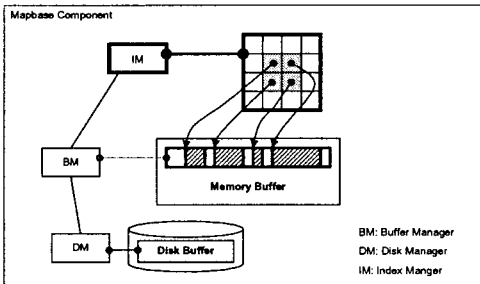


그림 6 성능 개선을 위한 인덱스관리자

Mapbase 컴포넌트에서는 그림 6과 같이 고정 그리드 파일 형태의 공간 인덱스를 생성하고 관리한다. 이 논문에서 공간 인덱스는 OGIS 데이터제공자로부터 공간 데이터를 읽어올 때, 함께 생성되어야 하므로 생성 비용이 상대적으로 적은 고정 그리드 파일을 선택하였다.

4. 결론

개방형 GIS 구조에서는 서로 다른 GIS 서버에서, 서로 다른 데이터 모델에 따라 구축된 공간 데이터를 동일한 방법으로 접근하기 위한 표준 인터페이스를 제공한다. 표준 인터페이스는 상호 운용성을 지원하지만, 반면에 데이터 모델의 변환에 따른 데이터 접근 속도의 저하를 초래한다. 이 논문에서는 공간 데이터 접근 속도를 개선하기 위해 OGIS 데이터소비자를 위한 버퍼 관리 모듈을 설계하였다. 전체 레이어에 대해서 통합적인 메모리 관리와 공간 인덱스를 생성·관리함으로써 버퍼관리자의 데이터 접근 속도를 개선하였다. 설계된 버퍼 관리 모듈은 OGIS 데이터소비자인 Mapbase 컴포넌트에서 구현되었으며, 공간데이터의 화면출력에서 좋은 성능을 보였다. 향후, 성능 개선에 대한 정량적 실험이 요구되며, 동적 인덱스 생성 방법에 대한 추가의 연구가 필요하다.

참고문헌

- [1] A. Kemper and D. Kossmann, "Dual-Buffering Strategies in Object Bases," In Proceedings of the 20th Conference on Very Large Data Bases, September 1994
- [2] Brinkhoff T., Horn H., Kriegl H.-p., Schneider R., "A Storage and Access Architecture for Efficient Query Processing in Spatial Databases," Proc. 3rd Int. Symp. On Large Spatial Database, Singapore, pp. 357-376, 1993.
- [3] E. J. ONeil, P. E. ONeil, and G.Weikum, "The LRU-K Page Replacement Algorithm For Database Disk Buffering," In Proceedings of the ACM SIGMOD International Conference on Management of Data, May 1993
- [4] Gisbert Droege, Hans-Jorg Scheck, "Query-adaptive data space partitioning using variable-size storage clusters," Advances in Spatial Databases, pp. 337-356, Springer-Verlag, 1993.
- [5] Mark Palmer and Stanley B. Zdonik, "Fido: A Cache That Learns to Fetch," In Proceedings of the 17th International Conference on Very Large Data Bases, September 1991
- [6] Nievergelt J., Hinterberger H., Sevcik K.C., "The Grid File: An Adaptable, Symmetric Multikey File Structure," ACM Trans. On Database Systems, Vol. 9, No. 1, pp. 38-71, 1984
- [7] OpenGIS Consortium Inc., The OpenGIS Simple Feature Specification for OLE/COM Revision 1.1, 1999