

벡터기반 웹 지리정보서비스의 설계 및 구현

이용식⁰ 이윤원 유승원 김승현 차상균
서울대학교 전기, 컴퓨터공학부
{doogie, kelovon, mokmon, shkim, chask}@kdb.snu.ac.kr

Design and Implementation of a Vector-based Web Geographic Information Service

Yong-Shik Lee⁰ Yoon-Won Lee Seung-Won Yoo Seung-Hyun Kim Sang K. Cha
School of Electrical & Computer Engineering, Seoul National University

요 약

최근 웹을 통하여 지리정보를 제공하는 서비스가 점점 보편화되어 가고 있다. 그러나 대부분의 서비스는 서버에서 모든 질의를 처리하고 결과를 이미지형태로 보여주는 구조를 가지고 있어 서버에 많은 부담을 주고 제공할 수 있는 서비스에 제약이 있다. 벡터데이터 기반 서비스의 경우 이러한 문제를 해결할 수 있으나 상대적으로 높은 클라이언트 성능을 요구한다. 그러나 최근 클라이언트 시스템의 성능이 향상됨에 따라 벡터데이터를 사용하여 클라이언트측의 자원을 사용하는 구조가 의미있게 되었다. 본 논문은 벡터데이터를 기반으로 하여 웹을 통해 지리정보를 제공하는 시스템을 설계하고 구현하였다. 사용자에게 불필요한 정보를 제거하고 데이터 전송량을 줄이기 위하여 레벨에 따른 상세정도(LOD: Level Of Detail)를 지원하고, 클라이언트 캐시, 데이터 압축 전송 등의 방법을 사용한다. 또한, 객체전송을 위한 통신모듈을 자동으로 생성해주는 틀을 사용하여 개발시간을 줄이도록 하였다.

1. 서론

최근 인터넷의 사용이 대중화됨에 따라 인터넷을 통하여 지리정보 데이터를 제공하는 서비스가 점점 보편화되어 가고 있다[1][2]. 그러나, 대부분의 이들 서비스는 단순히 서버에서 생성한 지도 이미지를 제공하는 정도에만 그치고 있어 지리정보와 연계된 부가서비스를 제공하기가 어렵다는 단점이 있다. 또한 서버에서 질의 결과를 생성하는 모든 작업이 이루어지므로 서버에 많은 부담을 주기 때문에 높은 서버 용량을 필요로 한다. 따라서, 이러한 문제점들을 해결하기 위하여 벡터데이터를 사용하여 서버에서 받아온 데이터를 클라이언트에서 처리하여 제공하는 방식이 개발되기 시작하였으며, 이에 대한 상용화가 이루어지고 있다[3].

벡터데이터를 사용한 시스템은 다음과 같은 장점이 있다. 첫번째는 데이터를 객체별로 관리할 수 있다는 것이다. 이를 통해 클라이언트에서 데이터를 저장하여 질의의 결과로 사용할 수가 있다. 둘째는, 데이터의 재사용이 용이하다는 것이다. 예를 들어, 지리정보 서비스의 기본기능인 확대, 축소 등의 기능을 구현하는데 있어 이미지 기반 서비스보다 유연성을 제공한다. 셋째는 서버에서 수행하던 데이터의 처리 중 일부를 클라이언트에서 처리하므로, 서버의 부담을 덜어 줄 수가 있다. 그리고, 벡터데이터를 사용하는 국제표준이 생겨나고 있어 앞으로 많은 연

구개발이 기대된다[4].

한편, 벡터데이터를 사용한 시스템은 이미지만을 받아보는 시스템에 비해서 상대적으로 높은 클라이언트 성능을 필요로 하고, 더 많은 양의 데이터를 전송해야 하는 경우가 생긴다. 그러나, 현재 초고속 통신망의 보급, 클라이언트 컴퓨터 성능의 향상으로 인해서, 이러한 문제점들이 점차 줄어들게 되었다. 따라서, 벡터데이터 기반의 웹 지리정보서비스가 점점 의미있게 되었고, 이에 대한 연구들이 진행되고 있다.[5]

본 시스템은 벡터데이터를 사용하여 웹기반 지리정보서비스에 사용될 수 있도록 설계되었다. 효율적인 서버 구조를 위하여 쓰레드 풀 구조를 사용하고, 레벨에 따른 상세정도(LOD:Level Of Detail)를 지원한다. 또한 서버의 소켓 통신모듈을 개발하는 부담을 줄이기 위하여 통신모듈 자동생성의 방법을 사용한다. 그리고, 통신비용을 줄이기 위하여 클라이언트 캐시, 데이터 압축 전송 등의 방법을 사용한다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 논문에서 제시하는 시스템 구조에 대하여 설명하고, 3 장에서는 구현된 시스템의 구현환경과 LOD, 통신모듈 자동생성, 데이터 압축전송, 클라이언트 캐시에 대하여 기술한다. 4 장에서 요약 및 향후 연구 계획으로 결론을 대신한다.

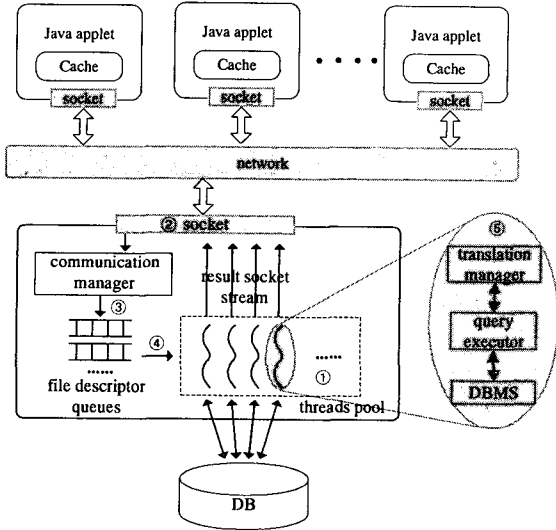


그림 1: 시스템 구조 및 동작과정

2. 시스템 구조

2.1 서버구조 및 동작과정

전체 시스템 구조는 그림 1 과 같다. 서버는 다수의 사용자의 요청을 효율적으로 처리하기 위하여 쓰레드 풀 구조로 설계하였다. 이 구조의 서버가 동작하는 과정은 다음과 같다.

- ① 서버가 실행되면, 쓰레드들을 생성시키고 각각의 쓰레드에 대해서 DBMS의 트랜잭션을 초기화 하고, 데이터 베이스를 연다. 이 쓰레드들이 풀을 형성한다.
- ② 소켓을 열고 클라이언트로부터의 요청을 기다린다.
- ③ 클라이언트로부터 요청이 들어오면 communication manager는 이 소켓의 file descriptor를 큐에 넣는다.
- ④ 쓰레드 풀에 있는 쓰레드들은 큐에 있는 file descriptor들을 꺼내서 translation manager 를 통해 소켓 스트림형태의 데이터들을 파싱한 후, 각각의 해당 요청들을 데이터베이스에 질의하여 처리한다.
- ⑤ translation manager 는 이 질의 결과를 다시 소켓 스트림으로 재구성한 후 클라이언트로 전송해 준다.
- ⑥ 각 쓰레드는 2~5의 과정을 반복한다.

2.2 클라이언트 및 GUI

클라이언트는 자바 애플릿으로 구현하여 기능의 확장이 용이하고 사용자의 플랫폼에 독립적이며 사용자와의 인터랙티브한 서비스의 구현이 쉽도록 하였다. 또한, 애플릿의 크기를 줄이기 위하여 자바 스크립트와의 연동, 바이트코드 줄임 등의 방법을 사용하였다. 이 방법을 통하여 애플릿 크기를 50% 이상 줄일 수 있었다.

그림 2는 사용자 인터페이스를 보여준다. 화면을 통하여서 마우스 입력을 받아 “확대이동”, “축소이동”, “중심이동” 등을 수행할 수 있고, 각종 컨트롤을 이용한 텍스트 질의가 가능하다. 또한 벡터데이터를 기반으로 한 애플릿 구조로 이루어져 있기 때문에 건물에 대한 하이퍼링크 등이 가능하고, 각 객체가 종류별로 레이어를 이루고 있어



그림 2: 사용자 인터페이스

특정 레이어를 선택하는 기능도 구현하였다. 나아가 각 객체에 대한 온라인 수정 등의 기능도 구현할 수 있다.

3. 시스템 구현

3.1 구현환경

본 시스템은 DBMS 로는 상용 객체지향 DBMS 인 Objectivity/DB[6]를 사용하였고, 서버는 Sun Enterprise 3000 위에서 SPARC C++ 컴파일러로 구현하였다. 클라이언트는 JDK1.1 을 사용하여 애플릿으로 구현하였다. 그리고, 비공간 DBMS 인 Objectivity/DB에서 지리 정보 데이터를 다루기 위하여 본 연구실에서 개발한 SDBX(Spatial DataBase extension), 공간 데이터 타입/연산자, 공간 인덱스 라이브러리를 활용하였다.

3.2 LOD(Level Of Detail)

데이터베이스는 효율적인 데이터 관리를 위하여 행정경계, 도로, 강, 지하철, 건물 등의 각 레이어별로 구분하여 저장한다. 이때, 실제 클라이언트에서 필요한 최소한의 데이터만을 저장하여 사용자에게 필요한 정보만을 제공하며 전송되는 데이터의 크기를 줄이기 위하여 레벨에 따른 상세정도(LOD: Level Of Detail)를 구현하였다[7]. LOD는 각 줌레벨 별로 사용자가 인식할 수 있는 최소한의 데이터만을 저장하기 위해 레벨에 맞게 간략화시키는 방법을 사용하여 구현하였다[8].

3.3 통신모듈 자동생성

서버가 데이터를 전송하기 위해서는 프로토콜이 필요 하다. 본 시스템 구조에서는 CORBA 와 같은 객체 기반 통신언어를 사용하여 구현할 수도 있으나, 대량의 데이터 전송으로는 적합하지 못하다[9]. 따라서, 성능향상을 위해 자체적으로 소켓을 이용한 프로토콜을 정의하여 구현하였다. 또한, 소켓 통신을 사용하는 경우 각 객체를 스트림으로 변환하여 전송해야 하는데, 많은 종류의 객체에 대하여 이러한 과정을 반복해야 하고, 새로운 객체종류가 생길 때마다 새로 프로그램을 작성해야 하는 문제가 있다. 이를 해결하

```

module AdminArea {
  struct Point {
    unsigned short x;
    unsigned short y;
  };
  struct AdminArea {
    unsigned short type;
    vector<Point> geometry;
    string name;
  };
  interface QueryProc {
    vector<AdminArea> get_AdminArea_gu(in long
    level, in long xl, in long yl, in long xh, in long yh);
  };
};
    
```

그림 3: ODL로 기술된 스키마

기 위하여 ODMG(Object Database Management Group)의 스키마 정의 언어인 ODL(Object Definition Language)형태로 기술된 스키마로부터 전처리하여 통신을 담당하는 모듈을 자동으로 생성하도록 하였다[10]. 그림 3은 ODL로 기술된 스키마의 예를 보여준다. 이 스키마를 전처리하면 Point객체와 AdminArea객체를 전송할 수 있는 통신모듈을 생성해 준다. 그림 4는 통신모듈 자동생성 과정을 나타내고 있다.

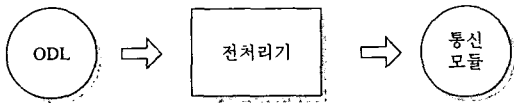


그림 4: 통신모듈 자동생성

3.4 클라이언트 캐시

클라이언트에서 데이터 재전송량을 최소화하기 위하여 캐시를 구현하였다. 캐시는 고정 그리드 방법과 해시 테이블을 이용하여 인덱스를 만들고, R-tree에 데이터를 저장하는 구조를 사용한다. 이를 통해 사용자가 한번 방문한 곳은 기억하여서 다시 같은 영역을 질의 할 때, 데이터를 다시 받아 와야 하는 비율을 감소시키도록 하였다. 또한 데이터가 LOD구조로 저장되어 있기 때문에 하위 레벨의 데이터를 상위레벨에서 다시 사용할 수 있는 장점이 있다.

4. 요약 및 향후계획

본 시스템은 벡터데이터를 사용하여 웹기반 지리정보서비스를 제공하는 효율적인 구조를 제안한다. 이를 위하여 LOD, 클라이언트 캐시, 데이터 압축 전송 등의 방법을 사용한다. 그리고, 통신모듈의 개발부담을 줄이기 위하여 통신모듈 자동생성의 방법을 사용한다. 앞으로 사용자의 응답시간을 최소화하기 위한 점진적 벡터 전송 방법에 대한 연구와 디스크 기반 DBMS 대신 현재 본 연구실에서 연구하고 있는 주메모리 기반 DBMS를 사용한 서버구조에 대하여 연구를 수행할 계획이다.

통신모듈 자동생성의 방법을 사용한다. 앞으로 사용자의 응답시간을 최소화하기 위한 점진적 벡터 전송 방법에 대한 연구와 디스크 기반 DBMS 대신 현재 본 연구실에서 연구하고 있는 주메모리 기반 DBMS를 사용한 서버구조에 대하여 연구를 수행할 계획이다.

5. 참고문헌

- [1] FreeMap, "http://www.freemap.co.kr".
- [2] MapQuest, "http://www.mapquest.com".
- [3] CyberMap, "http://www.cybermap.co.kr".
- [4] W3C, "Scalable Vector Graphics (SVG)," <http://www.w3.org/Graphics/SVG/Overview.htm> 8.
- [5] M. Bertolotto et al, "Progressive Vector Transmission", in *ACM GIS*, pp. 152-157, 1999.
- [6] Objectivity, Inc. , "Using Objectivity/C++ Supplement", 1998.
- [7] Sabine Timpf, "Abstraction, Levels of Detail, and Hierarchies in Map Series", in *Proc. of COSIT*, pp. 125-139, 1999.
- [8] D. H. Douglas, T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a line or its caricature," in *The Canadian Cartographer*, 10(2), pp. 112-122, 1973.
- [9] A. S. Gokhale, D. C. Schmidt, "Measuring the Performance of Communication Middleware on High-Speed Networks," in *SIGCOMM*, pp 306-317, 1996
- [10] Sang K. Cha, Kihong Kim, Changbin Song, Yongsik Kwon, Sangyong Hwang, "Efficient Web-Based Access to Multiple Geographic Databases through Automatically Generated Wrappers," in *Web Information System Engineering*, pp. 30-37, 2000.