이동객체의 위치정보 저장 시스템

Moving Object Location Information Storage System

장인성*, 박종현
In-Sung Jang*, Jong-Hyun Park

한국전자통신연구원 공간정보기술센터

e-mail:{e4dol2,jhp}@etri.re.kr

요 약

이동통신기술의 발전과 무선인터넷 사용자의 급증 및 휴대 단말기 장치의 성능 향상으로 인하여 사용자의 위치 정보를 활용하여 부가 정보 서비스를 받을 수 있는 위치기반서비스(LBS : Location Based Service)에 대한 관심이 급증하고 있다. 위치기반 서비스를 제공하기 위해서는 사용자의 위치정보를 관리하는 이동객체 관리 기술이 필수적으로 요구된다. 이에 본 논문에서는 기존의 시공간데이터베이스시스템 연구와 이동객체 관리 시스템을 간략히 살펴보고, 이동객체 객체 관리 시스템 중 이동객체를 저장하는 서브시스템을 설계 및 구현하고자 한다. 이를 위해서 단일 RDBMS 기반에 위치정보를 저장, 검색하는 컴포넌트와 수많은 이동객체의 계속적인 저장 및 검색의 과부하를 조절하고 대용량의 위치정보를 효율적으로 관리하기 위해 분산 데이터베이스에 저장하는 컴포넌트를 제안하고자 한다.

1. 서 론

이동통신기술의 발전과 무선인터넷 사용자의 급증 및 휴대 단말기 장치의 성능 향상으로 인하여, 최근 LBS는 무선 인터넷의 핵심 응용으로 주목 받고 있으며 이를 위한 기반 기술 확보가 민간 및 정부 차원에서 활발히 진행 중에 있다.

LBS의 정의에는 약간의 차이가 있지만, "휴대폰, PDA, 노트북 PC등 휴대용 단말기를 기반으로 사람이나 사물의 위치를 정확하게 파악하고, 그 위치와 관련된 부가 정보를 서비스 및 응용시스템"을 통칭한다. 3GPP(3rd Generation Partnership Project) TS 22.071에서는 "위치서비스는 위치기반의 응용제공이 가능한 네트워크를 이용한 표준화된 서비스"라고 정의하고 있으며, OGC(Open GIS Consortium)에서는 "위치기반 서비스는 위치 정보의 접속, 제공 또는 위치정보에 의해 작용하는 모든 응용 소프트웨어 서비스이다"라고 정의한다. 미국의 통신위원회에서는 "위치기반 서비스는 이동식 사용자가 그들의 지리학적 위치, 소재 또는 알려진 존재에 대한 서비스를 받도록 하는 것이다"라고 정의하고 있다.[1]

LBS의 기반기술은 크게 비즈니스 모델, 게이트웨이 기술, 위치측위 기술, LBS 플랫폼 기술 그리고 LBS 핵심 컴포넌트로 나누어지며, 이중 LBS 플랫폼 개발 시에 가장 중요한 요소 기술은 수많은 이동객체를 효과적으로 관리하는 이동객체 관리

기술이다.

이에 본 논문에서는 이동객체를 관리하는 기술에 관한 것으로 그 중에서도 이동객체를 저장하는 서브시스템을 제안하고자 한다. 본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로서 기존의 시공간 데이터베이스 연구를 살펴보고, 3장에서는 이동객체 관리 시스템의 전체 구성을 간략히 소개한다. 그리고 4장과 5장에서 이동객체를 저장하는 서브시스템으로서 단순저장 컴포넌트와 분산저장 컴포넌트에 대하여 설명한다. 마지막으로, 6장에서 결론을 맺겠다.

2. 관련연구

시공간 데이터베이스 분야는 시간에 따라 변화되는 공간정보를 어떻게 효율적으로 저장하고 검색하여 활용할 것인지에 대한 연구에 초점이 맞추어지고 있기 때문에 보다 역동적으로 변화하는 현상을 모델화하고, 변화의 시점을 파악할 수 있으며, 모든 데이터를 보유할 수 있는 특성을 갖고 있다. 기존의 연구는 유럽의 CHOROCHRONOS[2],미국의 NCGIA[3], 각 국의 시간데이터베이스 전문학자들이 참여하는 국제 센터인 TimeCenter[4] 등을 중심으로 연구되고 있다.

2.1 CHOROCHRONOS

CHOROCHRONOS는 유럽 시공간 데이터베이스 연구자들이 협력하여 1996년부터 수년간 집중적으로 추진된 학계중심의 공동연구 프로젝트이다. 주요연구 내용은 시공간 데이터베이스 (STDBMS, Spatio-temporal Database Systems) 구현을 목적으로 설계 및 아키텍처 관한 연구를 주로 수행하고 있다. 주요 연구 분야는 다음과 같다.

·공간과 시간의 분류법, 구조 및 표현
·STDBMS를 위한 모델 및 언어
·시공간 정보를 위한 그래픽사용자인터페이스
·시공간데이터베이스에서의 질의처리절차
·시공간데이터베이스의 저장구조 및 색인기술
·STDBMS 아키텍쳐

2.2 NCGIA

미국과학재단(National Science Foundation)의 후원 하에 "GIS를 이용한 지리정보의 기초적 연구"를 위한 학계중심의 연구센터이다. "GIS를 이용한 지리정보의 기초적 연구"를 수행하며, 구현보다는 구현을 위한 개념적 수준에서의 연구를 주로 추진하고 있다. CHOROCHRONOS에 비하여 다소 개념적이고 인지적 수준에서의 연구내용이다.

2.3 TimeCenter

덴마크, 미국을 비롯하여 우리나라 등 세계 여러 나라의 데이터베이스 관련학자들이 참여하는 시간 데이터베이스기술 개발을 위한 국제센터이다. 다양한 기존 그리고 새로운 DBMS기술을 기반으로 시간 질의언어 및 시간 데이터모델 등 시간 데이터베이스 기술의 이론적 기반을 마련하고 프로토타입를 개발하고자 하는 것이다. 주요연구 내용은 시간데이터베이스 중심의 연구센터로 시간데이터모델, 질의, 병렬처리, 색인, 이동객체 등의 연구가 활발히 추진되고 있으며, 프로토타입의 연구개발 성과를 도출하고 있다.

3. 이동객체 관리 시스템

이동객체 관리 시스템의 구성은 그림 1과 같다. 시스템은 위치 획득 게이트웨이 컴포넌트, 위치 질의 컴포넌트, 위치 저장 컴포넌트, 현재위치 색인 컴포넌트, 과거위치 색인 컴포넌트 그리고 응용프로그램으로 구성된다.

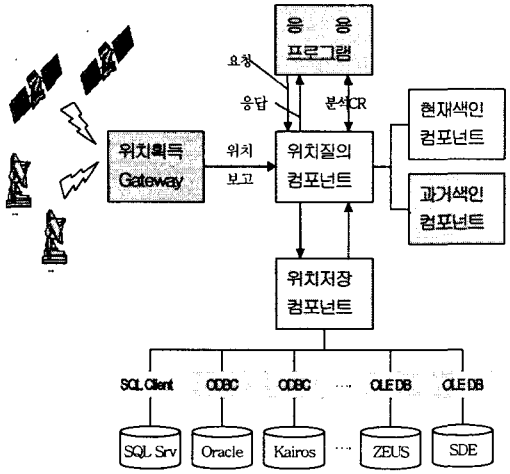

그림 1. 이동객체 관리 시스템

위치획득 게이트웨이는 인공위성(예: GPS), 이동 통신망 등 다양한 위치 측위 기술을 통해서 측위 된 위치정보를 제공한다.

위치질의 컴포넌트는 위치획득 게이트웨이는 통해 보고되는 이동객체의 위치정보를 저장하거나 응용프로그램이 요청하는 위치정보를 제공한다. 질의 처리 시 대용량의 위치 정보 빠르게 제공하기 위해, 이동객체의 현재위치를 관리하는 현재색인 컴포넌트와 과거 이동 궤적에 대한 과거색인 컴포넌트가 있다.

위치저장 컴포넌트는 4장과 5장에서 다시 살펴보겠다.

4. 단순 위치 저장 컴포넌트

위치 저장 컴포넌트는 위치질의 컴포넌트의 저장 및 검색을 처리하는 기능을 담당한다. 그림 2와 같이 단순저장 컴포넌트는 .NET 데이터 제공자를 통해서 위치정보를 저장하고 관리하게 된다. NET 데이터 제공자는 크게 3가지 타입이 있는데, MSSQL을 지원하는 타입과 OLE DB를 지원하는 타입 그리고 ODBC를 지원하는 타입이다. 따라서 OLE DB 인터페이스 또는 ODBC를 지원하는 데이터베이스는 그 종류에 상관없이 바로 위치정보의 저장소로 사용할 수 있다.

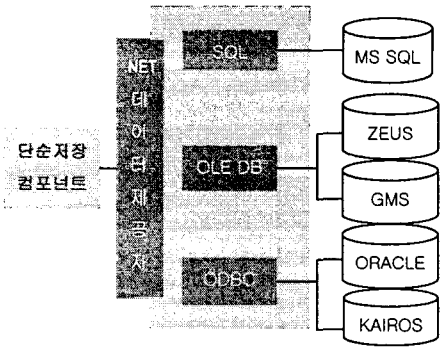

그림 2. 단순 저장 컴포넌트

4.1 테이블 스키마

이동객체를 저장할 때 테이블의 스키마는 두 가지 형태가 있다. 표 1-a은 위치정보가 보고될 때마다 바로 테이블에 저장하는 구조이다.

필드명	타입	예	비고
MOID	Long	0123456789	객체 식별자
X	Double	38492.32	X좌표
Y	Double	56431.52	Y좌표
MOTime	DateTime	"2003-09-15 11:02:32"	시간
MOError	Float	100	오차범위

a. 단순 테이블 스키마

필드명		타입	예	비고
MOID		Long	0123456789	객체식별자
MBR	MinX	Double	31000.00	최소 x 좌표
	MaxX	Double	55000.00	최대 x 좌표
	MinY	Double	41000.00	최소 y 좌표
	MaxY	Double	66431.52	최대 y 좌표
FromTime		DateTime	"2003-09-15 11:02:32"	시작시간
ToTime		DateTime	"2003-09-16 12:02:32"	종료시간
DataLenght		UInt	20	데이터 길이
Locations		BLOB or String	0XF23F3D123CB5.. "31001,35412.9,20,"	위치정보

b. 복합 테이블 스키마

표 1. 테이블 스키마

여기서, MOID는 이동객체의 식별자로 이동 단말기의 고유번호이다. MOTIME은 위치정보를 획득한 시간이고, MOERROR는 위치 정확도의 오차범위이다.

표 1-a의 테이블 구조로 저장하게 되면, 버퍼링등이 필요 없으므로 저장이 단순하다는 장점이 있다. 하지만, 위치정보가 보고 될 때마다 트랜잭션이 발생하고, 테이블의 Row의 수가 증가하여 성능이 감소할 뿐 아니라 특정 MOID의 이동 경로를 파악하거나 데이터를 분석할 때도 비효율적이다. 그래서 이동객체의 위치 정보를 MOID별로 일정량 동안 버퍼링을 한 후 해당 버퍼가 차면 저장하는 표 1-b과 같은 테이블 스키마를 사용한다.

Locations는 버퍼링된 이동경로를 저장하는 컬럼으로, BLOB(Binary Large OBject)을 지원하는 데이터베이스에는 BLOB(image,varbinary)형태로 저장하고 그렇지 않으면 문자열(varchar)로 저장한다.

4.2 저장 프로시저

이동질의 컴포넌트에서 삽입 및 검색 요청이 들어오면, 요청된 작업을 수행하기 위해 컴퓨터가 사용하는 명령어 세트는 작업이 수행되는 플랫폼에서 사용될 수 있는 가장 낮은 수준까지 분류된다. 속도를 위해, 요청이 있을 때마다 실행되는 질의와 데이터베이스 유지 작업을 생성하도록 할 수도 있지만, 이럴 경우에는 데이터베이스 서버에서 필요 이상의 많은 작업을 해야만 한다. 이를 막기 위해서 저장 프로시저를 사용하면 캐싱된 실행계획을 사용하기 때문에 성능의 향상을 기대할 수 있다. 표 2 저장 컴포넌트에서 저장 프로시저 중 삽입을 위한 저장 프로시저이다.

```
( @id int,          @length   int, @x1 float,
@y1 float, @x2 float, @y2 float,@fromTime
datetime, @toTime       datetime,  @locations
image )
AS BEGIN
insert into TABLENAME values(@id, @length,
@x1, @y1,@x2, @y2, @fromTime, @toTime,
@locations )
RETURN END
```

표 2. 삽입을 위한 저장 프로시저

5. 분산 위치 저장 컴포넌트

만약 이동객체 관리 시스템이 백만 명의 가입자를 관리하고, 10분에 한번씩 위치를 보고한다면, 1분당 십만 건의 이동객체 저장을 처리해야 한다. 단일 데이터베이스 시스템으로 감당하기 힘들기 때문에 분산 위치 저장 컴포넌트를 제안하고자 한다.

분산 위치저장 컴포넌트의 내부적으로 사용되는 테이블 스키마와 저장 프로시저는 4절과 동일하다.

분산 위치저장 컴포넌트는 서버관리자, 인덱스관리자 그리고 원격서버로 구성된다. 그림 3과 4는 분산위치 저장 컴포넌트에서의 저장과 검색에 대한 서브 컴포넌트간의 상관관계이다. 먼저, 그림 3에서 저장에 관한 흐름을 살펴보면, 먼저 사용하고자 하는 데이터베이스를 서버관리자에게 등록한다. 그 다음 위치 질의 컴포넌트가 서버관리자에게 삽입할 서버를 요청해서 결과를 받은 후 저장에 대한 메타정보는 인덱스 관리자에게 저장하고, 위치정보는 원격서버를 통해 해당 데이터베이스에 저장한다.

그림 4에서처럼 검색에 대해 살펴보면 다음과 같다. 질의 컴포넌트가 인덱스관리자에게 검색결과를 가지고 있는 해당 원격서버 목록을 가지고 와서 서버관리자에서 원격서버를 요청하고 원격서버에 접속에서 질의를 수행한다.

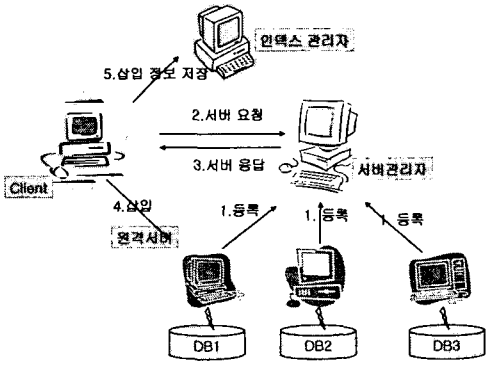

그림 3. 분산 위치 저장 컴포넌트의 저장

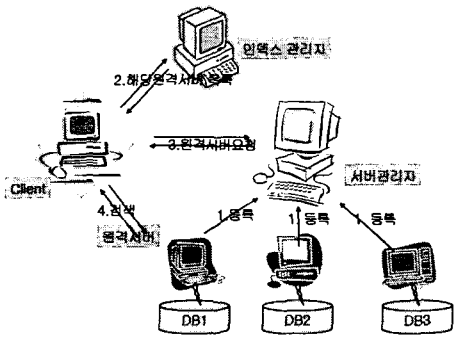

그림 4. 분산 위치 저장 컴포넌트의 검색

서버관리자는 다음과 같은 기능을 지원하는 모듈로 구성되어 있다. 원격서버를 등록하고 해지하는 서버등록모듈, 서버 해지 모듈 그리고 서버의 과부하를 조절하는 로드 밸런싱 모듈과 서버선택 모듈, 기타 기능으로 서버 리스트 반환 모듈과 디스크 상태 체크 모듈이다.

데이터를 분산 저장시키는 전략에는 다음과 같은 다양한 방법이 있다.

- 시간속성을 이용한 방법
  시간대별로 분산저장
- 가중치를 이용한 방법
  네트워크 부하,저장 공간 등을 고려하여 분산저장
- 접속 횟수를 이용한 방법
  최소 접속한 횟수를 이용하여 분산저장
- 가중치를 이용한 최소접속 횟수
  접속횟수와 네트워크부하,저장공간을 고려하여 분산
- 이동객체 식별자(MOID)를 이용한 방법
- 위에 방법들을 혼합하여 분산 저장하는 방법

본 논문에서는 가중치를 이용하여 분산 관리하는 방법을 사용한다. i번째 원격서버의 가중치(w_i)를 계산하는 방법은 아래 식과 같다.

$$w_i = \frac{n}{DB_i\text{에 } n\text{개의 Row를 저장하는 시간}}$$

이렇게 계산된 w_i의 최대 공약수 값이 실제 가중치이다. 가중치를 이용한 방법을 선택한 이유는 단위 시간처리 내에는 네트웍크의 성능, 서버의 CPU, 메모리에 따른 성능을 모두 포함하고 있기 때문이다. 그림6의 예를 들어 설명하면 다음과 같다. 데이터베이스 원격서버로 DB_1, DB_2, DB_3가 있고 각각의 서버에 1000개의 Row를 저장할 때 소요되는 시간이 1초: 2초:4초였다면 각각 서버의 가중치는 1000: 500: 250이 되고, 이 값의 최대 공약수를 구하면 4,2,1이 된다.

처음 DB_1, DB_2, DB_3의 가중치는 4,2,1이다. Row 저장 질의가 요청되면 가중치가 가장 높은 DB_1의 가중치를 3으로 감소 시키고 DB_1에 저장한다. 그 다음 저장 질의에 대해서도 DB_1의 가중치가 3으로 가장 높으므로 2로 감소시킨다. 그 다음 질의에 대해서 DB_1, DB_2가 가중치 2로 같을 때는 최근에 사용되지 않은 DB_2를 선택한다. 이렇게 반복을 수행한 후 모든 가중치가 0이 되면 초기값을 재설정해서 반복을 수행해서 분산시키는 전략이다.

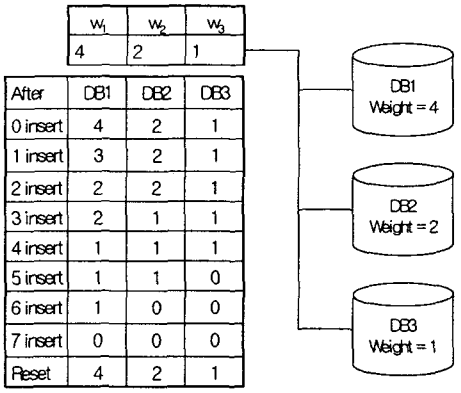

그림 6. 분산 저장 방법.

6. 결론

최근 들어, 사용자의 현재 위치를 정확하게 파악하고, 그 위치와 관련된 여러 가지 부가정보 서비스를 제공하는 LBS에 대한 관심이 급증하게 되면서, 사용자의 현재 위치뿐만 아니라 과거 위치를 효과적으로 저장하고 검색할 수 있는 이동 객체 관리 시스템이 요구된다. 이에, 이동

객체의 수가 많을수록, 위치 갱신 시간이 짧을수록 이동객체의 정보는 대용량이 되는데 이를 효과적으로 저장 및 검색하기 위한 위치저장 컴포넌트를 설계하는 것이다. 이를 위해 단순 위치 저장 컴포넌트와 분산전략을 통한 분산저장 컴포넌트를 설계 및 구현했다. 분산 전략에 다양한 방법이 있겠지만 본 논문에서 서버 당 처리 시간을 이용한 가중치기반에 분산 전략을 사용하였다.

참 고 문 헌

[1]SoftBank Research, IT Insight Strategy Report,"LBS, Now & Future"

[2]http://www.dbnet.ece.ntua.gr/~choros/index.html

[3]http://www.ncgia.ucsb.edu/

[4]http://www.cs.auc.dk/TimeCenter/about.htm