

# TMO 기반 분산 이동 객체 데이터베이스의 설계 및 구현

이준우<sup>0</sup> 전세길 나연목

단국대학교 컴퓨터공학과

{joonwoo<sup>0</sup>, sgjeon}@dankook.ac.kr, ymnah@dku.edu

## A Design and Implementation of TMO-based Distributed Moving Object Database

Joonwoo Lee,<sup>0</sup> Segil Jeon, Yunmook Nah

Department of Computer Engineering, Dankook University

### 요약

시공간 데이터베이스에서 저장대상이 되는 이동객체는 실시간으로 위치가 변화하는 동적인 데이터이다. 동일한 객체의 위치경로가 여러 데이터베이스 서버에 분산 저장 될 수 있으며, 질의 역시 여러 서버에 걸쳐서 분산질의 형태로 수행될 필요성이 있다. 이를 위해서 본 논문에서는 실시간 객체 모델인 TMO(Time-triggered Message-triggered Object)를 이용하여 이동 객체를 발생시키고 객체의 위치를 저장, 질의하는 시스템을 설계 및 구현하였다. TMO를 이용한 분산 저장을 통해 입력 시 발생할 수 있는 오버헤드를 줄이고, 데이터베이스간에 완전 연결 네트워크가 형성되어 각 데이터베이스 서버간의 상호 작용을 최소화 할 수 있게 된다. 이 시스템은 실시간으로 분산 위치정보를 관리해야 하는 여러 응용 분야에서 효과적으로 활용될 수 있다.

### 1. 서론

최근 들어, 항공기 운항 제어, 지능적 교통제어, 이동통신 시스템 등 많은 응용 분야에서 계속적으로 움직이는 객체 데이터에 대한 저장 및 검색의 필요성이 증가하고 있다.

현재 많이 이용하고 있는 위치 정보서비스의 경우에서 보면 계속적으로 변화하는 객체 데이터를 데이터베이스에 유지해야 하는 경우가 있다. 예를 들어, 어떤 특정지역을 관찰하는 순찰차의 경우 순찰차의 유지 및 관리 차원에서 이동경로, 현재위치, 예상 경로등에 대한 정보가 필요하게 된다. 이러한 경우 순찰차의 위치정보가 계속해서 데이터베이스에 저장되어야 한다.

계속적으로 발생하는 이동객체의 위치 저장은 데이터 특성상 실시간으로 이루어져야 한다. 이러한 이동객체의 특성으로 인해 시간에 의해 구동되는 메소드의 활성화와 데드라인을 강화할 수 있는 기능을 제공하는 실시간 객체 모델이 필요하다. 본 논문에서는 효율성과 신뢰성이 높은 실시간 객체 모델인 TMO (Time-triggered Message-triggered Object) 모델을 사용한다[1, 2].

이동객체의 저장을 위한 시스템 구조로는 모든 데이터가 한곳에 저장되어야 하는 중앙 단일 시스템 구조와 각 데이터가 여러 서버에 걸쳐 저장되는 분산 시스템 구조가 있다. 중앙 단일 시스템 구조에서는 특정 서버의 많은 연산으로 인한 성능 저하를 초래 할 수 있기 때문에 위치 정보를 객체의 현재 위치를 기반으로 여러 개의 데이터베이스 서버 시스템에 분산 저장하는 것이 적합하다. 본 논문에서는 위치 정보의 분산 저장 및 처리를 위한 시스템을 TMO를 기반으로 설계 및 구현한다.

질의처리는 각 DB와 연결된 TMO에서 처리된 후 결

과물을 전송하고 종합해서 보여준다.

구현 시스템에서 사용되는 시뮬레이션 데이터는 TMO의 시간 구동 방식의 메소드를 이용하여 일정 시간마다 위치 정보를 발생시켜 사용하며, 질의 처리 시 분산된 데이터베이스 서버간의 통신은 메시지 구동 방식으로 수행하게 된다.

본 논문의 구성은 다음과 같다. 제2장에서는 분산데이터베이스 시스템과 실시간 객체 모델인 TMO에 대해서 살펴보고, 3장에서는 시스템의 설계와 저장 및 질의처리 흐름에 대해서 살펴본다. 4장에서는 시스템의 구현에 대해 설명을 한다. 5장에서는 결론을 맺고 향후 연구 과제를 제시한다.

### 2. 관련연구

본 장에서는 분산데이터베이스 시스템에 대하여 소개하고, 본 논문에서 사용한 실시간 시스템인 TMO에 대하여 설명한다.

#### 2.1 분산 데이터베이스

물리적으로 한 장소에 데이터베이스가 설치, 운영되면 중앙 단일 시스템이다. 지리적으로 분산된 데이터베이스를 사용자가 마치 하나의 중앙 집중 데이터베이스 시스템과 같이 사용할 수가 있는 시스템을 분산 데이터베이스(distributed database)라 한다. 분산 데이터베이스에서는 데이터가 물리적으로 떨어진 여러 데이터베이스에 저장되며, 각 데이터베이스는 일반적으로 서로 독립적으로 운용될 수 있는 DBMS를 하나씩 가지고 있다. 분산 데이터베이스는 성능상의 이유뿐 아니라 데이터 소유의 지역성과 데이터 가용성 향상 등의 이유로 데이터를 분산시킨다[4, 5].

### 2.2 TMO 모델

TMO 모델은 기존의 객체 모델에 대한 확장으로써, 다양한 정확도를 제공해 주고, 시간 구동 메소드를 갖고 있으며, 객체에 포함된 모든 메소드에 대한 데드라인을 강화하였다[1,2]. TMO 모델이 가지는 고유한 특징을 간단히 살펴보면 다음과 같다.

- 두 종류의 메소드: 메소드에 명시된 시간 조건에 의해 구동되는 SpM(Spontaneous Method)과 클라이언트로부터의 요청 메시지에 의해 구동되는 SvM(Service Method)의 두 종류가 있다.
- 객체의 데이터는 ODSS (Object Data Store Segment)에 저장되어 진다.
- BCC(Basic Concurrency Control): SpM과 SvM이 동시에 내부 데이터를 접근하였을 때에는 SpM이 우선 접근한다.
- TMO 객체에 포함된 SpM과 SvM은 모두 한계 시간을 갖는다.
- TMO는 독립적이며 서로 다른 TMO 간에 통신이 자유롭다.

### 3. 시스템 설계

#### 3.1 구조

각 데이터베이스는 하나의 TMO와 연결되어 있고 TMO 간의 통신이 자유롭게 이루어지기 때문에 완전 네트워크로 구성된 분산 데이터베이스가 된다. 그래서 완전 네트워크로 구성된 다른 분산 데이터베이스와 달리 설치비용이 적게 들지만 최대의 신뢰성과 융통성을 제공 받게 된다.

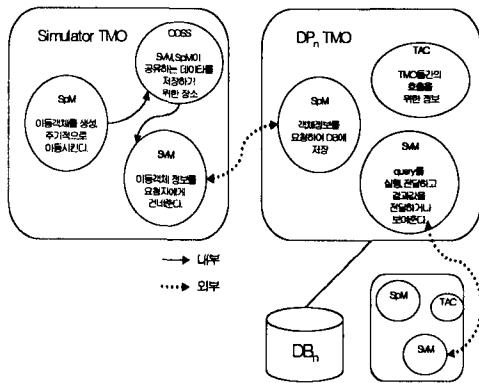


그림 1. TMO 모델을 이용한 모델링

그림 1에서 Simulator TMO는 이동객체를 생성하고 생성된 이동객체가 움직게 해주고 DP<sub>n</sub>은 SpM을 통해 주기적으로 자신의 영역에 포함되는 데이터를 Simulator TMO의 SvM에 요청하게 되고 전달받은 데이터를 DB에 저장한다.

각 DP는 하나의 TMO로 이루어진다. 질의처리를 위해 각 DP는 SvM을 통해 서로 질의문을 전달하고 데이터를 전송한다.

### 3.2 저장 시나리오

그림 2는 객체 o1이 시간 t1부터 t9사이에 움직인 경로와 이때 테이블에 기록되는 내용이다. DP1 지역내에서 발생한 정보는 DB1에 저장되고 DP1 지역에서 벗어나 DP2로 이동하게 되면 DB2로 저장된다. C++로 구현된 TMO를 ODBC로 MS SQL 2000과 연결하여 데이터를 저장한다. Table 구조는 표1과 같다.

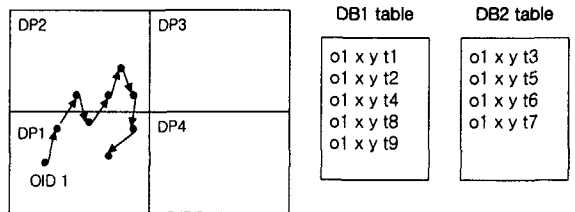


그림 2. 이동객체 저장 시나리오

표 1. Table 구조

name	자료형	설명
OID	int	객체 ID
x_co	int	x 좌표
y_co	int	y 좌표
time	datetime	시간

### 3.3 질의처리

분산 데이터베이스의 목표중 하나는 위치 투명성 (location transparency)을 제공하는 것이다. 본 논문에서는 이러한 위치 투명성을 위해 질의처리시 다른 사이트의 데이터에 대한 요청은 다른 DB에서 처리하여 결과를 질의가 들어온 곳으로 가지고 오는 방법을 택하였다

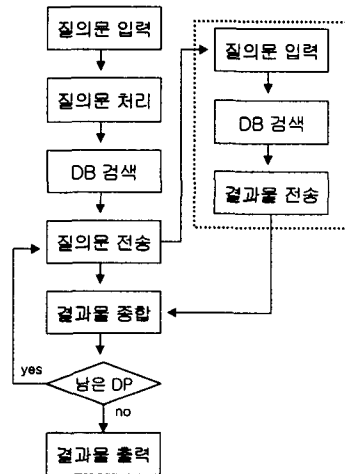


그림 3. 질의처리 흐름도

그림 3은 질의처리 흐름도에 대해 표현한 것이다. 어 한 DP에서 질의가 들어올 경우 질의를 받은 DP는 질의를 수행하고 다른 DP에게 질의문을 전달하고, 질의

를 받은 DP는 질의문을 수행하여 그 결과를 질의를 보낸 DP에 전달한다. 질의문이 입력된 DP는 수행한 결과와 전달 받은 결과문을 종합하여 출력하게 된다.

### 3.4 분산 데이터베이스의 주요 함수

본 논문에서 사용한 TMO 는 C++ API로 코딩되어 있다. 주요함수의 구현은 C++ API를 이용하였으며 다음의 그림과 같다.

```

DBCConnect()
DBDisconnect()
DBExecuteSQL()
Produce(int num)
Move()
GetObject(Object, count, area)
    
```

그림 4. 분산 데이터베이스에서 사용된 함수

DBCConnect와 DBDisconnect 함수는 TMO와 데이터베이스간 연결에 사용된다. ODBC로 연결되어 있기 때문에 DSN 이름, 사용자 ID, 비밀번호등을 필요로 한다.

본 논문의 구현은 이동객체 데이터를 저장만 하기때문에 DBExecuteSQL()은 simulator로부터 받은 데이터를 저장하기 위한 insert문 만을 실행하게 된다.

Produce()는 이동객체를 생성한다. Move()는 생성된 객체를 일정한 범위내의 임의 위치로 이동시킨다. GetObject()는 해당 DP에 포함되는 객체를 배열에 포함하여 전달한다.

### 4. 구현

본 논문에서 제안한 분산 데이터베이스의 구현 환경으로 하드웨어는 Pentium III 550MHz, 메모리 256MB에 OS는 Windows 2000을 사용하는 두 대의 컴퓨터를 사용하였다. 상용 DBMS는 MS SQL Server 2000을 이용하였다.

그림 5는 발생되어진 이동객체가 해당 데이터베이스에 저장되는 것을 보여주는 화면이다.



그림 5. TMO 구동 및 DB 입력 화면

그림 6은 입력한 질의에 따른 결과를 보여주고 있다.



그림 6. 질의처리 결과창

### 5. 결론

본 논문에서는 이동객체를 발생시키고 실시간으로 분산 저장 처리는 시스템을 실시간 객체 모델인 TMO를 사용해 설계, 구현하였다. 또한 분산 데이터베이스의 중요한 특징중 하나인 위치 투명성을 제공하기 위해 질의 처리시에 관련 질의를 해당 데이터베이스 서버에서 처리하여 결과를 가지고 취합하는 방법을 사용하였다.

본 논문에서 제안한 시스템은 계속적으로 변화하는 위치 정보를 효율적인 방법으로 실시간 저장, 처리할 수 있다.

본 논문의 향후 연구 과제로는 분산데이터베이스 시스템의 가용성을 위해 데이터의 중복을 허용하도록 시스템을 변경하고 성능을 비교 분석하는 연구등을 들 수 있다.

### 참고문헌

- [1] Kim, K.H., Ishida Masaki and Liu Juqiang, "An Efficient Middleware Architecture Supporting Time-Triggered Message-Triggered Objects and an NT-based Implementation," Proc. 2nd IEEE CS Int'l Symp. on Object-Oriented Real-Time Distributed Computing (ISORC'99), St.Malo, France, May 1999, pp.54-63.
- [2] Kim, K.H., "Object-Oriented Real-Time Distributed Programming and Support Middleware," Proc. ICPADS2000(7th Int'l Conf. on Parallel & Distributed Systems), IEEE CS Press Iwate, Japan, July 2000, pp.10-20.
- [3] Kane Kim, *TMO Support Library(TMOSL)*, UCI DREAM Lab, 2002.
- [4] 이석호, 데이터베이스 시스템, 정익사, 2000.
- [5] Silberschatz, Korth and Sudarshan, *DATABASE SYSTEM CONCEPTS*, McGrawHill, 1997.