

MODBMS를 위한 분산 위치 저장 컴포넌트의 설계 및 구현[†] (Design and Implementation of a Distributed Location Storage Component for MODMBS)

윤재관*, 장유정*, 장인성**, 한기준*

(Jae-Kwan Yun*, Yu-Jung Jang*, In-Sung Jang**, Ki-Joon Han*)

*건국대학교 컴퓨터공학과, **한국전자통신연구원

*{jkyun, yjjang, kjhan}@db.konkuk.ac.kr, **e4dol2@etri.re.kr

초록

최근 들어 사용자의 위치 데이터를 기반으로 한 LBS가 유용한 서비스로 등장함에 따라 이동 객체의 관리에서 발생하는 대용량의 위치 데이터에 대한 처리가 중요한 문제가 되었다. 이러한 대용량의 위치 데이터는 초당 몇 백만 건이 생성 및 변경되기 때문에 위치 데이터에 대한 신속한 저장 및 검색을 위한 시스템이 필요하다.

이에 본 논문에서는 위치 데이터를 효과적으로 분산 저장 및 검색할 수 있는 MODBMS를 위한 분산 위치 저장 컴포넌트를 설계 및 구현하였다. 분산 위치 저장 컴포넌트는 미들웨어의 정보를 관리하는 서버 관리자, 검색할 서버의 정보를 반환하는 연결 관리자, 디스크 정보 및 백업 작업을 위한 디스크 관리자, 위치 데이터의 저장 및 검색을 위한 미들웨어, 그리고 클라이언트로 구성되어 있다.

1. 서론

최근 들어 무선 인터넷과 모바일 장치의 발전으로 인하여 사용자의 위치 데이터를

이용하는 위치 기반 서비스(LBS: Location Based Service)가 유용한 서비스로 등장하고 있다[1,4,6,9]. 이러한 위치 데이터는 GPS를 통하여 얻어지며, 위치 데이터를 처리하는 서버에서는 초당 몇 십만 건에서 몇 백만 건의 데이터 처리가 필요하다[10]. 그러므로, 이러한 대용량의 위치 데이터를 단일 시스템, 단일 데이터베이스에 저장하는 것은 시스템의 성능이 매우 뛰어나더라도 데이터 처리에 상당한 시간을 소모하게 되므로 비효율적이다[8,11].

이와 같은 단점을 극복하기 위해서는 처리해야 하는 위치 데이터의 양을 줄이거나 여러 시스템에 분산시켜 작은 단위로 처리하는 것이 가장 효율적이다[7]. 그러므로, 이와 같이 대용량의 위치 데이터를 여러 서비스에서 효과적으로 사용하기 위해서는 현재 및 과거의 위치 데이터를 다양한 데이터베이스 시스템에 효과적으로 분산하여 저장하고 검색할 수 있는 분산 위치 저장 컴포넌트가 필요하다.

본 논문에서는 이동 객체에 대한 위치 데이터를 효과적으로 분산 저장 및 검색할 수 있는 이동 객체 데이터베이스 시스템을 위한 분산 위치 저장 컴포넌트를 설계 및 구현하였다. 본 논문에서는 많은 분야에서 사

[†] 본 연구는 ETRI의 공간 정보(4S) 연계 기술 지원 사업에서 지원받았음.

용되고 있는 데이터베이스 시스템인 MS-SQL Server, 오라클, ZEUS를 이용하여 미들웨어를 개발하였고, 미들웨어를 관리하기 위한 서버 관리자, 위치 데이터 검색을 위한 연결 관리자, 저장 공간 관리를 위한 디스크 관리자, 그리고 클라이언트를 개발하였다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로써 .NET 리모팅과 MODBMS에 대하여 소개한다. 그리고, 3장과 4장에서는 각각 분산 위치 저장 컴포넌트의 설계 및 구현에 대하여 설명한다. 마지막으로, 5장에서 결론에 대하여 언급한다.

2. 관련 연구

본 장에서는 분산 위치 저장 컴포넌트를 개발하기 위해서 사용된 .NET 리모팅과 MODBMS에 관하여 설명한다.

2.1 .NET 리모팅

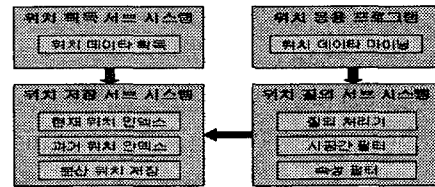
.NET 프레임워크는 SOAP, XML, HTTP와 같은 표준 프로토콜을 사용하는 네트워크를 통해서 프로그래밍 방식으로 웹 응용 프로그램, 클라이언트 응용 프로그램, XML 관련 응용 프로그램, Web Service 응용 프로그램 등을 작성, 구축 및 실행할 수 있게 해주는 .NET 플랫폼의 프로그래밍 모델이다[3]. .NET 프레임워크는 기존의 방대한 리소스를 차세대 응용 프로그램과 서비스에 통합하고, 인터넷 범위 응용 프로그램의 구축 및 작업 문제를 빠르게 해결할 수 있는 생산성 높은 표준 기반 환경을 제공한다.

.NET 리모팅은 .NET 프레임워크를 통해서 다른 응용 프로그램 도메인에 있는 객체들이 서로 상호 작용할 수 있도록 해준다

[5]. 또한, .NET 리모팅은 원격 응용 프로그램에서 메시지를 주고받는 통신 채널뿐만 아니라 활성화 및 수명 지원을 포함한 다양한 서비스를 제공한다.

2.2 MODBMS

MODBMS는 한국전자통신연구원(ETRI)에서 LBS와 같은 서비스에서 이동 객체의 위치 데이터를 효과적으로 관리하기 위하여 개발된 시스템이다. MODBMS는 크게 위치 획득 서버 시스템, 위치 저장 서버 시스템, 위치 질의 서버 시스템, 그리고 위치 응용 프로그램으로 구성된다. MODBMS의 전체 구조는 <그림 1>과 같다.



<그림 1> MODBMS의 전체 구조

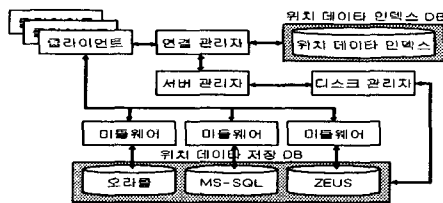
위치 획득 서버 시스템은 GPS를 이용하여 위치 데이터를 획득하고, 위치 저장 서버 시스템은 이동체의 현재 위치, 과거 위치에 대한 인덱스를 생성하며, 발생한 대용량의 위치 데이터를 분산하여 저장한다. 위치 질의 서버 시스템은 위치 저장 서버 시스템에 질의를 전달하며 시공간 필터, 속성 필터 등의 기능을 제공한다. 위치 응용 프로그램은 위치 데이터를 이용하여 위치 데이터 마이닝과 같은 작업을 수행한다. 본 논문에서의 분산 위치 저장 컴포넌트는 MODBMS의 위치 저장 서버 시스템 중 분산 위치 저장의 기능을 담당한다.

3. 시스템의 설계

본 장에서는 MODBMS를 위한 분산 위치 저장 컴포넌트의 설계에 대하여 설명한다.

3.1 시스템의 구조

본 논문에서 개발한 분산 위치 저장 컴포넌트의 주된 기능은 .NET 환경에서 네트워크를 기반으로 하여 대용량의 위치 데이터를 저장하고 검색하는 것이며, 이를 위해서 <그림 2>와 같은 시스템 구조를 갖고 있다.



<그림 2> 시스템 구조도

분산 위치 저장 컴포넌트는 크게 서버 관리자, 연결 관리자, 디스크 관리자, 미들웨어, 클라이언트로 구성된다. 그리고, DB는 위치 데이터 저장 DB와 위치 데이터 인덱스 DB로 구성된다. 위치 데이터 저장 DB는 이동 객체에서 발생하는 대용량의 위치 데이터를 저장하며, MS-SQL Server, 오라클, ZEUS가 사용된다. 위치 데이터 인덱스 DB는 위치 데이터를 빠르게 검색하기 위하여 사용되며 연결 관리자에 의해서 인덱스 데이터가 저장 및 검색된다.

위치 데이터를 검색하기 위해서는 우선 클라이언트가 연결 관리자에게 검색하고자 하는 위치 데이터의 정보를 질의하고, 반환된 위치 데이터의 미들웨어 정보를 이용하여 해당하는 미들웨어에 접속한다. 미들웨어는 위치 데이터 저장 DB에 질의를 전달하고 위치 데이터의 레코드 셋을 클라이언트로 반환하게 된다.

위치 데이터를 저장하기 위해서는 우선

클라이언트가 서버 관리자에게 위치 데이터를 저장할 수 있는 위치 데이터 저장 DB를 할당받는다. 그리고 나서, 해당하는 위치 데이터 저장 DB의 미들웨어에게 위치 데이터를 저장하기 위한 질의를 전달하고, 성공적으로 저장이 되었을 경우 연결 관리자에게 저장된 미들웨어의 정보를 전달하게 된다. 만약 위치 데이터의 저장이 실패하였을 경우에는 실패한 데이터에 대한 로그를 남긴다.

디스크 관리자는 서버 관리자에 등록된 미들웨어의 위치 데이터 저장 DB 정보를 수시로 검사하면서 데이터의 저장 시에 디스크 공간이 부족한지를 확인하고 만약 공간이 모자라면 서버 관리자가 더 이상 데이터를 저장하지 않도록 설정한 후에 해당하는 위치 데이터 저장 DB의 상태를 읽기 모드로 변경한다. 또한, 디스크 관리자는 저장된 위치 데이터가 오래되어 자주 사용되지 않을 경우에 사용자의 요구에 의하여 자동으로 백업하는 기능을 수행한다.

3.2 위치 데이터의 타입

본 논문에서 사용되는 데이터의 타입은 MOID, MOMBR, MOTime, MOTLocation, MORow가 있다. MOID는 저장되는 위치 데이터를 식별하기 위한 ID로써 System.UInt32를 사용하고, MOMBR은 저장되는 위치 데이터 전체에 대한 MBR 값을 설정하기 위한 데이터 타입으로 Left, Right, Top, Bottom 값이 저장된다. MOTime은 위치 데이터가 생성된 시작 날짜와 끝 날짜를 저장하기 위한 데이터 타입으로 System.DateTime을 사용한다.

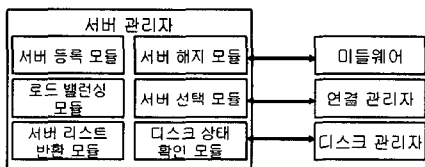
MOTLocation은 하나의 위치 데이터 값을 설정하기 위한 데이터 타입으로 시간, X좌표 값, Y좌표 값, 오류 값이 저장되는 데이터 타입이다. MORow는 한번에 저장해

야 하는 위치 데이터 셋의 최소 단위로써 MORow 타입은 우선 위치 데이터의 ID, 전체 포함된 MOLocation의 개수 값인 Length, MBR의 값, 그리고 위치 데이터가 생성되기 시작한 시간과 종료된 시간을 가진다. 이후에는 Length 값만큼의 MOLocation 데이터가 들어가게 된다.

MORow 타입은 MOLocation의 개수에 따라 길이가 결정되는 가변 길이 데이터이기 때문에 위치 데이터 저장 DB에서는 이를 처리하기 위하여 LONG RAW, IMAGE와 같은 가변 길이 데이터 타입이 사용된다.

3.3 서버 관리자의 설계

서버 관리자는 분산된 환경에서 이동 객체의 위치 데이터를 저장하기 위한 미들웨어의 정보를 유지 관리하며 새로운 위치 데이터를 삽입할 때 로드 밸런싱을 수행한다. 본 논문에서의 서버 관리자는 <그림 3>과 같이 서버 등록 모듈, 서버 해지 모듈, 로드 밸런싱 모듈, 서버 선택 모듈, 서버 리스트 반환 모듈, 디스크 상태 확인 모듈로 구성되어 있다.



<그림 3> 서버 관리자의 구조

서버 등록 모듈과 서버 해지 모듈은 각각 미들웨어가 처음 시작될 때 서버 관리자에게 미들웨어의 IP, PORT 번호, 서비스 이름을 등록하고, 미들웨어가 종료될 경우에 등록을 해지하는 모듈이다. 로드 밸런싱 모듈은 새로운 위치 데이터의 저장이 요구될 경우에 유효한 미들웨어를 선택하기 위한 모듈이다. 서버 선택 모듈은 연결 관리자에

서 호출되며 로드 밸런싱 모듈에서 선택된 서버의 정보를 반환하는 역할을 한다. 서버 리스트 반환 모듈은 모든 미들웨어의 리스트를 반환한다. 디스크 상태 확인 모듈은 미들웨어의 리스트 정보를 이용하여 각 위치 데이터 저장 DB의 상태를 확인하고 그 결과를 기록한다.

3.4 연결 관리자의 설계

연결 관리자는 위치 데이터의 인덱스가 저장되어 있는 위치 데이터 인덱스 DB를 관리하며 이동 객체의 위치 데이터에 대한 삽입 및 검색이 일어날 경우 위치 데이터 인덱스 DB를 이용하여 데이터를 검색하고, 삽입이 일어날 경우 위치 데이터의 인덱스 내용을 위치 데이터 인덱스 DB에 저장한다.

위치 데이터 인덱스 DB의 스키마는 <표 1>과 같이 구성된다. 본 논문에서는 위치 데이터 인덱스 DB의 관리를 위해 MS-SQL Server를 사용하고 있으며 레이어별로 MORow가 분산 저장될 때 미리 위치 데이터 인덱스 DB에 레이어 이름, MOID, 시작 시간과 끝 시간을 저장한다. 그리고, 위치 데이터 인덱스 DB에는 MORow가 저장된 미들웨어의 서버 IP, PORT 번호, 그리고 서비스 이름이 추가로 등록된다.

<표 1> 위치 데이터 인덱스 DB 스키마

컬럼 이름	타입	길이	NULL	설명
mlayer	varchar	20	NO	레이어 이름
moid	int	4	NO	ID
fromtime	datetime	8	NO	시작 시간
totime	datetime	8	NO	종료 시간
serverip	varchar	20	NO	미들웨어의 IP
serverport	int	4	NO	미들웨어의 PORT
servicename	varchar	20	NO	미들웨어의 서비스 이름
servicetype	varchar	6	NO	MSSQL, ORACLE, ZELUS

위치 데이터를 검색할 경우 클라이언트는 연결 관리자로부터 해당하는 위치 데이터가 있는 서버 정보를 얻어오게 된다. 클라이언트는 이 서버 정보를 이용하여 데이

타가 저장되어 있는 미들웨어들을 순회하면서 위치 데이터를 검색한다. 연결 관리자에서 검색되는 데이터의 결과는 레이어별, MOID별로 정렬이 되어서 반환된다.

3.5 디스크 관리자의 설계

디스크 관리자는 서버 관리자에 등록된 미들웨어가 사용하는 위치 데이터 저장 DB의 디스크 상태를 주기적으로 확인하면서 공간이 부족할 경우에는 더 이상의 삽입이 이루어지지 않고 검색만 수행할 수 있도록 설정하는 기능을 수행한다. 그리고, 사용자에게 의하여 저장된 위치 데이터를 다른 위치 데이터 저장 DB로 백업하는 기능도 수행한다. 이러한 기능을 수행하기 위해서 디스크 관리자는 <그림 4>와 같은 구조가 사용된다.



<그림 4> 디스크 관리자의 구조도

디스크 관리자에는 서버 관리자를 통해서 등록된 미들웨어의 정보를 지속적으로 탐지하는 서버 정보 탐지 모듈과 서버가 더 이상 위치 데이터를 저장하기 어려운 경우에 서버의 상태를 읽기 모드로 변환하고 백업이 일어난 후에 다시 쓰기 모드로 변환하기 위한 서버 상태 변환 모듈이 있다.

사용 공간 확인 모듈과 여유 공간 확인 모듈은 지속적으로 등록된 서버의 디스크 상태를 검사하여 사용하고 있는 디스크 공간과 사용할 수 있는 디스크 공간을 확인한다. 그리고, 백업 모듈은 사용자의 요청에 의하여 등록된 다른 미들웨어로 전체 데이터를 백업하는 기능을 수행하며, 레코드 개수 확인 모듈은 각 미들웨어마다 몇 개의

레코드가 저장되어 있는지를 검사한다.

3.6 미들웨어의 설계

미들웨어는 여러 데이터베이스 시스템과 연결하여 서버의 상태를 지속적으로 검사하고 MORow 데이터의 저장 및 검색 기능을 수행한다. 위치 데이터를 저장하기 위한 스키마는 MORow의 위치 데이터를 저장하기 위하여 설계되었으며 위치 데이터 저장 DB가 지원하는 데이터 타입에 따라 대용량의 위치 데이터가 바이너리 형태나 문자열의 형태로 저장된다.

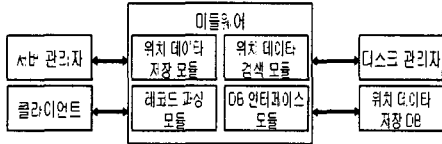
본 논문에서는 MS-SQL Server와 오라클 경우에는 IMAGE, LONG RAW와 같은 바이너리 데이터 타입을 사용하고, ZEUS 경우에는 문자열의 형태를 사용한다. 일반적으로 바이너리 형태로 데이터를 처리하는 것이 문자열을 파싱하여 데이터를 처리하는 것보다 성능이 좋고 디스크 공간을 적게 차지한다. <표 2>는 MS-SQL Server에서 위치 데이터 저장을 위해 사용되는 스키마를 보여준다.

<표 2> 위치 데이터 저장 DB 스키마

컬럼 이름	타입	길이	NULL	설명
mayer	varchar	20	NO	레이어 이름
moid	int	4	NO	ID
molelength	int	4	NO	위치 GIOI의 개수
monbr_x	datetime	8	YES	moloc에 저장되는 위치 데이터를 포함하는 MER의 값
monbr_y	varchar	8	YES	
monbr_xz	int	8	YES	
monbr_yz	varchar	8	YES	
molfrom	datetime	8	YES	상성된 시간
molto	datetime	8	YES	종료된 시간
moloc	image	16	YES	바이너리 형태로 변환된 전체 위치 데이터를 저장

미들웨어는 해당하는 위치 데이터 저장 DB와 연결하여 위치 데이터를 저장 및 검색하는 기능을 담당하는데, <그림 5>는 미들웨어의 구조도를 나타낸다. 클라이언트는 연결 관리자를 통하여 해당하는 미들웨어의 .NET 리모팅 서비스에 연결한 다음 위치 데이터 저장 및 위치 데이터 검색 모듈에 질의 처리를 요구한다. 이 과정에서 MORow 타입을 바이너리 형태로 변환하거

나 바이너리 타입에서 MORow 타입으로 변환하기 위해서 레코드 파싱 모듈이 사용된다. DB 인터페이스 모듈은 ADO.NET을 이용하여 구현된다.



<그림 5> 미들웨어의 구조도

4. 시스템의 구현

본 장에서는 MODBMS를 위한 분산 위치 저장 컴포넌트의 구현에 대하여 설명한다.

4.1 구현 환경

본 논문에서 분산 위치 저장 컴포넌트를 구현하기 위하여 사용된 시스템 환경은 다음과 같다. 분산 위치 저장 컴포넌트가 개발된 운영체제는 Microsoft Windows XP이고, 구동되는 .NET 프레임워크는 Microsoft .NET Framework v1.0.3705가 사용되었다. 개발 도구는 Microsoft Visual Studio .NET을 이용되었으며, 개발 언어는 C#이 사용되었다.

4.2 서버 관리자의 구현

.NET 리모팅 환경에서 서버를 구동하기 위해서는 우선 클라이언트와 서버가 동시에 사용할 수 있는 함수가 구현된 DLL이 필요하다. 서버 관리자 DLL은 서버 관리자의 기능이 구현된 DLL이며 <그림 6>과 같은 함수가 있다.

```
public IEnumerator GetServerList()
public int AddServer(string ip, int port, string type, int hitvalue)
public void DeleteServer(string ip, int port)
public ServerInfo GetServer()
```

<그림 6> 서버 관리자의 함수

GetServerList 함수는 등록된 서버의 정보를 알고자 할 때 호출되는 함수로서 등록된 미들웨어의 IP, PORT 번호, 서비스 이름 정보를 포함하고 있는 ArrayList를 반환한다. AddServer 함수는 미들웨어의 IP, PORT 번호, 서비스 이름을 이용하여 서버 관리자에 미들웨어를 등록하는 함수이며, hitvalue는 미들웨어가 처리하고 있는 작업의 양을 나타내며 별다른 요청이 없을 경우 내부적으로 자동 증가 및 감소된다. DeleteServer 함수는 등록되어 있는 미들웨어가 등록을 해제할 경우에 사용되는 함수이고, GetServer 함수는 클라이언트에서 데이터의 저장이 요청될 경우에 로드 밸런싱 과정을 거쳐 가장 작업이 적은 서버를 선택하는 함수이다.

4.3 연결 관리자의 구현

연결 관리자에서 사용되는 함수는 그림 7과 같이 위치 데이터를 저장할 경우 위치 인덱스 DB에 데이터를 삽입하기 위한 InsertIntoDb 함수와 위치 데이터를 검색할 경우 위치 인덱스 DB에서 데이터를 검색하는 SearchFromDb 함수가 있다. InsertIntoDb 함수의 인자는 레이어 이름, MOID, 시작 시간, 끝 시간, 서버 IP, 서버 PORT, 서비스 이름이고, SearchFromDb 함수에는 검색하고자 하는 레이어 이름, MOID 리스트, 시작 시간, 끝 시간이 인자로 사용된다.

```
public bool InsertIntoDb(string LayerName, MOID id,
MOTime fromTime, MOTime toTime, string serip, int port,
string service)
public IEnumerator SearchFromDb(string layerName,
MOID[] molDs, MOTime fromTime, MOTime toTime)
```

<그림 7> 연결 관리자의 함수

4.4 디스크 관리자의 구현

디스크 관리자에서 사용되는 함수는 그림 8과 같이 사용자의 요구에 의하여 위치 데이터를 백업하기 위한 MakeBackup 함수, 서버의 사용 공간, 여유 공간, 레코드 개수 등을 확인하기 위한 GetServerStatus 함수, 그리고 위치 데이터 저장 DB의 상태를 변환하기 위한 ChangeDBStatus 함수가 있다.

```
public bool MakeBackup(ServerInfo source, ServerInfo Dest)
public ServerStatus GetServerStatus(ServerInfo dest)
public bool ChangeDBStatus(ServerInfo dest)
```

<그림 8> 디스크 관리자의 함수

4.5 미들웨어의 구현

미들웨어를 구현하기 위해서 사용되는 클래스는 크게 ETRICCommon 클래스로부터 상속을 받아 각 위치 데이터 저장 DB에 맞게 데이터의 저장 및 검색 함수를 구현한 ETRIMOSTorageDistribute 클래스와 MORow 타입의 위치 데이터를 바이트, 및 텍스트 형태로 변환하기 위한 ETRICConverter 클래스가 있다. 그림 9는 미들웨어 DLL의 ETRIMOSTorageDistribute 클래스에서 사용되는 함수를 보여준다.

```
public bool InsertRow(string layerName, MOTRow moRow)
public MOTRow ParseMoRow(SqlDataReader myDataReader)
public override IEnumerator Search(string layerName, MOID[]
molDs, MOTime fromTime, MOTime toTime)
public override IEnumerator Search( string layerName,
MOID[] molDs, MOPeriod moPeriod)
public override void Packaging()
```

<그림 9> ETRIMOSTorageDistribute 함수

InsertRow 함수는 레이어 이름과

MORow 데이터 타입을 연속된 바이트로 변환한 MOTRow를 인자로 사용하여 위치 데이터 저장 DB에 이동 객체의 위치 데이터를 저장하는 함수이다. ParseMoRow 함수는 위치 데이터 저장 DB로부터 검색된 위치 데이터를 다시 MORow 위치 데이터 타입으로 변환하기 위해서 사용된다. Search 함수는 레이어 이름, MORow의 리스트, 시작 시간, 끝 시간을 인자로 하여 위치 데이터 저장 DB로부터 일치되는 위치 데이터를 검색하는 기능을 한다. Packaging 함수는 일련의 위치 데이터를 하나의 Package 형태로 그룹을 형성하는 기능을 한다.

<그림 10>은 ETRICConverter 클래스의 함수를 보여준다. .NET 리모팅에서는 클라이언트와 서버가 서로 데이터를 전송하기 위하여 이진 포맷기를 사용하는데 ETRICConverter 클래스는 이진 스트림에 모든 데이터를 순차화(Serialize)하는 기능을 수행한다.

```
public DateTime ByteToDateTime(byte[] buffer)
public byte[] DateTimeToByte(DateTime date)
private byte[] StringToByteArray(String str1)
private String ByteArrayToString(byte[] BA1)
public byte[] MOMBRTToByte(MOMBR mbr)
public byte[] MOTLocationToByte(MOTRow row)
public byte[] Serialize()
public MOTRow Deserialize(byte[] buffer)
```

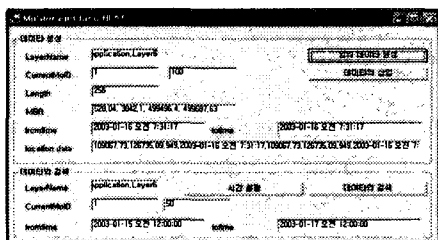
<그림 10> ETRICConverter 함수

ByteToDateTime, DateTimeToByte 함수는 DateTime 타입의 데이터를 바이트 형태로 변환하거나 바이트 타입의 데이터를 DateTime 형태로 변환하기 위한 함수이다. StringToByteArray, ByteArrayToString 함수는 문자열을 바이트 형태로 변환하거나 바이트 형태의 데이터를 문자열로 바꾸는 함수이다. MOMBRTToByte는 MORow 데이터 타입 중 MOMBR 데이터 타입의 데이터를 바이트로 변환하는 함수이며, MOTLocationToByte 함수는 MOTLocation 데이터 타입의 데이터를 바이트로 변환하

기 위한 함수이다. Serialize 함수는 .NET 리모팅에서 클라이언트가 서버로 데이터를 전송하기 위하여 이진 포맷기를 사용하여 바이트 스트림으로 순차화하는 함수이며, Deserialize 함수는 순차화된 데이터를 원래의 MORow 데이터 타입으로 환원하는 함수이다.

4.6 클라이언트의 구현

본 논문에서는 구현된 분산 위치 저장 컴포넌트의 테스트를 위하여 <그림 11>과 같은 클라이언트 인터페이스를 구현하였다. 클라이언트는 여러 시스템에서 동시에 실행될 수 있으며 하나의 미들웨어에 동시에 질의를 수행할 수 있다.



<그림 11> 클라이언트 인터페이스

5. 결론

최근 들어 사용자의 위치 데이터를 기반으로 한 LBS가 유용한 서비스로 등장하고 있다. 이러한 위치 데이터는 대용량으로 생성 및 변경되기 때문에 이동 객체에 대한 위치 데이터의 신속한 저장 및 검색을 위한 시스템이 필요하다. 이에 본 논문에서는 이동 객체에 대한 위치 데이터를 효과적으로 분산 저장 및 검색할 수 있는 MODBMS를 위한 분산 위치 저장 컴포넌트를 개발하였다.

본 논문에서는 차세대 운영체제인 .NET

환경에서 C#을 이용하여 분산 위치 관리 컴포넌트를 개발하였다. 이를 위해, MS-SQL Server, 오라클, ZEUS에 대용량의 위치 데이터를 저장하는 미들웨어, 분산된 미들웨어의 정보를 관리하는 서버 관리자, 빠른 검색을 위하여 위치 데이터의 인덱스를 관리하며 해당 미들웨어에게 위치 데이터를 검색하는 질의를 보내는 기능을 하는 연결 관리자, 서버의 디스크 사용을 관리하는 디스크 관리자, 그리고 각 관리자 및 시스템이 정상적으로 수행되는지 테스트 및 성능 평가를 수행하는 클라이언트를 개발하였다.

본 논문에서 개발된 MODBMS를 위한 분산 위치 저장 컴포넌트는 대용량의 위치 데이터를 효과적으로 분산하여 저장하고 신속하게 검색할 수 있게 함으로써 LBS와 같은 대용량의 이동 객체의 위치 데이터를 사용하는 서비스에서 유용하게 사용될 수 있다. 향후에는 대용량의 위치 데이터를 보다 빠르게 검색하기 위하여 화일을 이용하는 방법과 대용량의 데이터를 효과적으로 압축하는 방법에 대한 연구가 필요하다.

참고문헌

- [1] ISO TC/211, *19132 Geographic information - Location based services possible standards*, <http://www.isotc211.org/scope.htm#19132>.
- [2] Lee, Y., and William, C., *Mobile Communications Design Fundamentals*, John Wiley & Sons, 1993.
- [3] Microsoft Corporation, "Overview of the .NET Framework," <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000451>, 2001.
- [4] Niedzwiadek, H., "OpenLS-1 Interoperability Project: An Overview," LBS 포럼 창립총회 및 기념세미나, 2002, pp.30-51.
- [5] Obermeyer, P., and Hawkins, J.,

"Microsoft .NET Remoting: A Technical Overview," <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/hawkremoting.asp>, 2001.

[6] OpenLS Initiative, *A Request for Technology In Support of an Open Location Services(OpenLS™) Testbed*, <http://www.openls.org>, 2000.

[7] Prakash, R., and Singhal, M., *A Dynamic Approach to Location Management in Mobile Computing Systems*, Dept. of Computer and Information Science, The Ohio State Univ., Technical Report, OSU-CISRC-4/96-TR22, 1996.

[8] 윤재관, 이근호, 한기준, "Shareware DBMS를 이용한 모바일 GIS의 개발," 개방형 GIS연구회 논문지, 2권2호, 2001, pp.25-37.

[9] 윤재관, 장영승, 한기준, "모바일 GIS를 위한 위치 기반 서비스," 한국정보과학회 데이터베이스 연구, 18권1호, 2002.3, pp.3-15.

[10] 진희채, 박상미, 안병익, "위치기반정보 서비스를 지원하는 시스템 구조 및 소프트웨어 기술동향 분석," 2001 개방형 지리정보시스템 학회 학술회의 논문집, 4권1호, 2001, pp.145-160.

[11] 최혜옥, "위치기반서비스 (LBS, Location-Based Services)," 제3회 공간정보 워크샵, 2002년, pp.5-22.