

Oracle 기반의 대용량 지도 서비스를 위한 XServer의 설계 및 구현†

Design and Implementation of the XServer for Oracle-based Mass Map Services

신중수*, 김동오, 강홍구, 박춘걸, 한기준
건국대학교 컴퓨터정보통신공학과

Jung-Su Shin*, Dong-Oh Kim, Hong-Koo Kang,
Chun-Geol Park, Ki-Joon Han
Dept. of Computer Information & Communication Engineering,
Konkuk University

요지 정보 사회가 발전하고 정보의 활용이 늘어남에 따라 공간 데이터가 다양한 분야에서 활용되고 있다. 그리고, 공간 데이터가 널리 활용됨에 따라 ESRI와 같은 다양한 지리 정보 시스템(Geography Information System)이 발전하게 되었다. 그러나, 기존의 지리 정보 시스템은 다양한 분야에서의 활용을 위해 많은 기능을 제공함으로써 인해 일반적으로 대용량 공간 데이터에 대한 검색이 비효율적이다. 특히, 네트워크 환경이 발전하고 컴퓨팅 파워가 증가함에 따라 점차 대용량의 지도 서비스를 제공하는 분야에서 기존의 지리 정보 시스템 적용 시 공간 데이터 검색 속도가 저하되는 문제가 발생한다.

따라서, 본 논문에서는 안정적인 상용 DBMS인 Oracle을 기반으로 대용량 공간 데이터를 효율적으로 검색할 수 있는 Oracle 기반의 대용량 지도 서비스를 위한 XServer를 설계 및 개발하였다. XServer는 다양한 클라이언트의 질의를 효율적으로 처리하기 위한 질의 처리 관리자, 대용량의 공간 데이터를 빠르게 검색하기 위한 공간 인덱스 관리자 및 데이터 버퍼 관리자, 대용량의 공간 데이터를 안정적으로 저장하기 위해서 Shape 파일에서 추출한 공간 데이터를 Oracle에 저장 및 관리하기 위한 수입/수출 관리자와 DB 관리자로 구성되어있다. 마지막으로, 본 연구에서 개발한 Oracle 기반의 대용량 지도 서비스를 위한 XServer와 Oracle Spatial을 비교함으로써 기능을 검증하고 성능의 우수함을 입증하였다.

1. 서 론

최근 공간 데이터 수집 수단의 획기적인 발전으로 대용량의 공간 데이터가 축적되고, 현대 사회가 정보 사회로 발전하면서 정보의 활용이 늘어남에 따라 공간 데이터가 다양한 분야에서 요구되고 활용되고 있다[3,9]. 그리

고, 공간 데이터가 다양한 비즈니스 분야에서 널리 활용됨에 따라 공간 데이터의 저장 및 처리하는 ESRI와 같은 다양한 지리 정보 시스템이 크게 발전하게 되었다.

그러나, 기존의 지리 정보 시스템은 다양한 분야에서의 공간 데이터 활용을 위해 많은 기능을 제공함으로써 인해 대용량 공간 데

† 본 연구는 (주)유삼씨엔씨의 산학연구사업의 연구 결과로 수행되었음.

이타에 대한 검색이 비효율적이다. 특히, 네트워크 환경이 비약적으로 발전하고 컴퓨팅 파워가 증가함에 따라 점차 대용량의 지도 서비스를 제공하는 분야에서 기존의 지리 정보 시스템 적용 시 공간 데이터 검색 속도가 저하되는 문제가 발생한다[2,8].

따라서, 본 논문에서는 안정적이고 확장성을 가지는 상용 DBMS인 Oracle을 기반으로 대용량 공간 데이터를 효율적으로 검색할 수 있는 Oracle 기반의 대용량 지도 서비스를 위한 XServer를 설계 및 개발하였다[10]. XServer는 다양한 클라이언트의 질의를 효율적으로 처리하기 위한 질의 처리 관리자, 대용량의 공간 데이터를 빠르게 검색하기 위한 공간 인덱스 관리자 및 데이터 버퍼 관리자, 대용량의 공간 데이터를 안정적으로 저장하기 위해서 Shape 화일에서 추출한 공간 데이터를 효율적으로 Oracle에 저장하기 위한 수입/수출 관리자와 DB 관리자로 구성되어 있다.

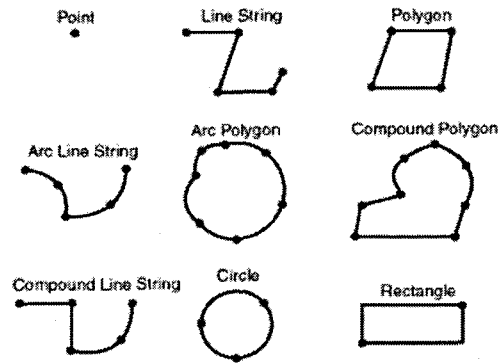
본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구로 Oracle Spatial과 OCCI (Oracle C++ Call Interface)에 대해 살펴본다. 제 3장과 제 4장에서는 대용량 지도 서비스를 위한 XServer의 설계 및 구현에 대해 상세히 설명하고, 제 5장에서는 성능 평가 결과를 살펴본다. 마지막으로, 제 6장에서는 결론을 언급한다.

2. 관련 연구

본 장에서는 XServer와 성능 비교 대상이 되는 Oracle Spatial을 살펴보고, Oracle과 XServer를 연동하기 위해 사용된 OCCI에 대해서 알아본다.

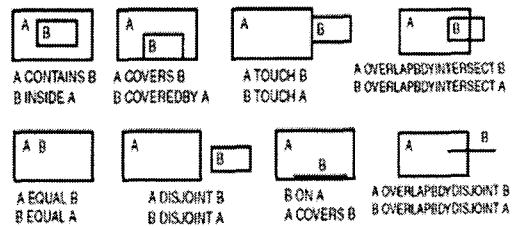
2.1 Oracle Spatial

Oracle Spatial은 Oracle에서 공간 데이터를 효과적으로 저장하고 빠르게 접근 및 분석할 수 있도록 하는 함수와 프로시저의 통합된 집합이다[2,5,7]. <그림 1>은 Oracle Spatial에서 지원하는 공간 데이터 타입을 보여준다.



<그림 1> Oracle Spatial 공간 데이터 타입

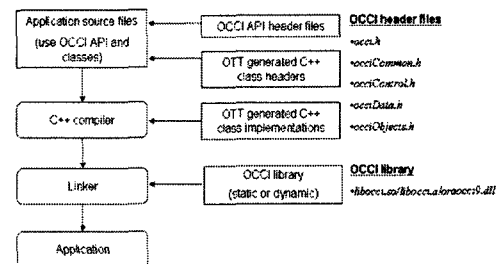
<그림 2>는 Oracle Spatial에서 지원하는 공간 연산을 보여준다. Oracle Spatial의 공간 질의는 공간 인덱스에서만 검색하는 SDO_FILTER 연산과 공간 인덱스 검색 결과를 바탕으로 실제 객체를 검색하는 SDO_RELATE 연산으로 구분된다. Oracle Spatial은 공간 인덱스로 Quadtree와 R-tree를 지원한다[1,2,4].



<그림 2> Oracle Spatial 공간 연산

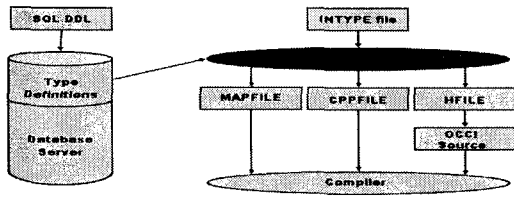
2.2 OCCI

OCCI는 Oracle에 접근하기 위한 C++ API로서, OCCI를 이용한 개발 과정은 그림 3과 같다.



<그림 3> OCCI 개발 과정

<그림 3>에서 보듯이 OCCI를 이용한 프로그램을 개발하기 위해서는 소스 파일 내에 OCCI API 헤더 파일과 OTT로 생성한 C++ 클래스 헤더 파일을 포함시킨다. 그리고 소스 파일을 컴파일하고 OCCI 라이브러리의 링크를 통해 어플리케이션을 생성한다. OTT는 Oracle에서 사용자 정의 타입(User-Defined Type)을 C++ 클래스로 변환해주는 프로그램이다. <그림 4>는 OTT의 실행 과정을 보여준다.



<그림 4> OTT 실행 과정

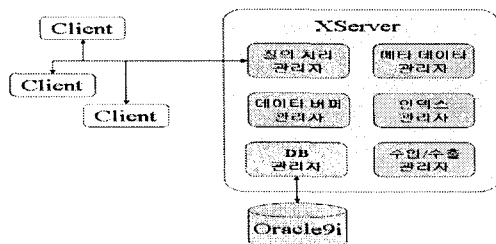
<그림 4>에서 보듯이 사용자 정의 타입을 정의하고 사용자 정의 타입에 대한 INTYPE 파일을 만들어 OTT를 구동시킨다. 다음 OTT 구동을 통해 생성되는 파일을 프로젝트에 포함시킨다[6].

3. 시스템 설계

본 장에서는 Oracle 기반의 대용량 지도 서비스를 위한 XServer의 시스템 설계에 대해서 상세하게 설명한다.

3.1 전체 시스템 구조

본 논문에서 설계한 Oracle 기반의 대용량 지도 서비스를 위한 XServer 시스템의 전체 시스템 구조는 <그림 5>와 같다.



<그림 5> 전체 시스템 구조

3.2 질의 처리 관리자

질의 처리 관리자는 클라이언트로부터 요청된 공간 및 비공간 데이터의 검색, 삽입, 갱신 질의를 처리하기 위한 모듈로서 인덱스 관리자, DB 관리자, 메타 데이터 관리자, 데이터 버퍼 관리자와 연동해서 요청된 질의를 처리한다. 질의 처리 관리자가 처리하는 검색 질의는 지도와 같은 공간 데이터에 대해 질의하는 공간 데이터 질의와 지역, 기관, 명칭 등과 같은 속성 데이터에 대해 질의하는 속성 데이터 질의로 나뉜다. 특히, 공간 데이터 질의 시에는 인덱스 관리자와 연동하여 관련된 인덱스에 대한 검색 및 갱신이 발생하게 된다. 질의 처리 관리자의 질의 명세는 <표 1>과 같다.

<표 1> 질의 명세

타입 ID	질의 타입	반환	인자
Q1	공간 영역 검색	공간 데이터	MBR, Schema_List, Table_List
Q2	객체 유형 검색	객체 속성 정보	Schema_Name, Table_Name, Object_List or *
Q3	레이어 검색	레이어 리스트	
Q4	공간 데이터 검색	객체 데이터 정보	Schema_Name, Table_Name
Q5	데이터 검색	영역/속성 정보	Schema_Name, Table_Name, Object_List, 공간데이터, 속성데이터
Q6	데이터 갱신	영역/속성 정보	Schema_Name, Table_Name, Object_List, 공간데이터, (속성데이터)
Q7	데이터 삭제	영역/속성 정보	Schema_Name, Table_Name, Object_List

3.3 데이터 버퍼 관리자

데이터 버퍼 관리자는 질의의 응답시간을 단축하기 위해 자주 사용되는 공간 데이터를 데이터 버퍼에 저장하고 검색하기 위한 모듈이다. 데이터 버퍼 관리자는 자주 사용되는 레이어를 버퍼에 저장하고 메타 데이터 관리자를 통해 버퍼 사용 유무 플래그와 버퍼 ID를 저장한다. 사용자의 질의 요청시 질의 처리 관리자가 데이터 버퍼 관리자의 인터페이스를 통해 데이터 버퍼에 접근하여 질의를 처리함으로써 Oracle에 접근할 필요없이 결과를 얻을 수 있어 빠른 질의 응답 시간을 보장할 수 있게 된다.

3.4 인덱스 관리자

인덱스 관리자는 공간 데이터를 효율적으로 검색하기 위해 공간 인덱스를 지원하는 모듈로서 인덱스 관리자에서 R-tree와 같은 공간 인덱스를 이용하여 Overlap, Contain,

Disjoint 등과 같은 다양한 공간 연산을 수행하게 된다. R-tree의 리프 노드는 오라클의 Rowid와 MBR 정보로 구성되며, 질의 요청시 인덱스 관리자는 인덱스 검색을 통해 검색된 Rowid 목록을 반환한다. 질의 처리 관리자는 반환된 Rowid 목록을 활용해 DB 관리자에 질의할 질의문을 생성하게 된다.

3.5 DB 관리자

DB 관리자는 Oracle에 대한 접속, 저장, 검색, 갱신 요청을 처리하는 모듈로서 OCCI를 이용해서 Oracle과 연결하여 저장, 검색, 갱신 요청을 효율적이고 안정적으로 처리한다. 또한 DB 관리자는 연결되는 세션의 수를 제한해서 세션의 수 증가로 인한 DB 성능 저하를 막는다.

3.6 메타 데이터 관리자

메타 데이터 관리자는 질의 처리시 참조할 데이터의 스키마로 이루어진 메타 데이터를 저장, 검색, 갱신, 관리하는 모듈이다. 스키마에는 레이어의 개수와 레이어 정보가 저장되며, 레이어 정보는 질의 처리시 참조하는 레이어 이름과 MBR 정보, 공간 데이터 타입 정보, 속성 데이터의 컬럼의 수와 타입 정보를 가진다.

메타 데이터 관리자에서 스키마는 수입/수출 관리자에서 Shape 파일 수입시에 분석한 Shape 파일 정보를 기반으로 생성되며, 질의 처리 관리자의 삽입, 삭제, 갱신의 질의 처리 과정에서 변경되는 메타 정보를 메타 데이터에 반영한다

3.7 수입/수출 관리자

수입/수출 관리자는 Shape 파일에서 공간 데이터를 추출하여 Oracle에 저장하기 위해 DB 관리자에 전달하고 DB 관리자를 이용해서 Oracle에서 공간 데이터를 추출하여 Shape 파일로 저장하는 모듈이다. Shape 파일은 지리 현상에 대한 기하학적 위치와 속성 정보를 저장하는 비위상(non-topological) 구조의 데이터 포맷을 지원한다. 수입/수출

관리자는 Shape 파일을 Oracle에 저장할 때 메타 데이터 정보, 공간 데이터, 속성 데이터를 구분해서 저장한다.

4. 시스템 구현

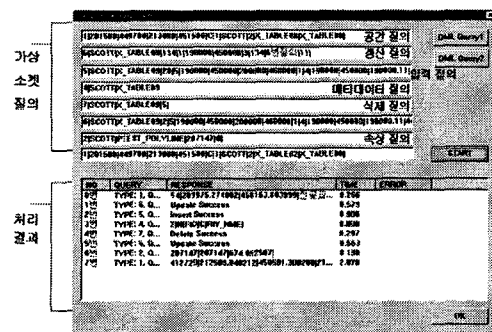
본 장에서는 Oracle 기반의 대용량 지도 서비스를 위한 XServer의 구현 환경을 살펴보고, 각 관리자의 구현에 대해 상세히 설명한다.

4.1 시스템 구현 환경

본 논문에서 XServer를 구현하기 위해서 운영체제는 Microsoft Windows XP SP2를 사용하였으며, 데이터베이스는 Oracle 9i (9.2.0.6) for Windows XP를 사용하였다. 개발 도구는 Microsoft Visual C++ 6.0 MFC를 사용하였고, Oracle과 연결하기 위한 OCCI와 사용자 정의 타입을 연동하기 위한 OTT를 사용하였다.

4.2 질의 처리 관리자

질의 처리 관리자는 질의를 분석하고 처리하여 결과를 반환한다. <그림 6>는 질의 처리 관리자를 수행하기 위해 사용한 질의 클라이언트 다이얼로그를 보여준다.



<그림 6> 질의 클라이언트 다이얼로그

<그림 6>에서 각 질의는 '!'를 구분자로 하는 문자열로 이루어져 있으며, 현재 공간 영역 질의, 객체 속성 질의, 메타 데이터 질의, 입력/갱신/삭제 질의가 질의 처리 관리자에 의해 동시에 수행되는 모습을 볼 수 있다.

<그림 6>의 첫 번째 질의는 <표 1>의 질의 명세에 따라 공간 질의임을 알 수 있다. 질의 처리 관리자는 명시된 MBR 정보, 공간 연산자 종류, 대상 스키마 및 레이어 정보를 이용해 메타 데이터 관리자로부터 해당 레이어의 인덱스 ID를 검색하고, 인덱스 관리자를 통해 MBR과 Contain 되는 후보 Rowid 목록을 받아 Oracle에 질의할 질의문을 생성한다. 질의문은 DB 관리자를 통해 Oracle로 질의하게 된다. 마지막으로 질의 처리 관리자가 질의 결과를 수집하여 반환한다.

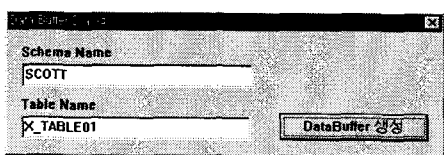
4.3 데이터 버퍼 관리자

데이터 버퍼 관리자는 자주 질의되는 레이어를 데이터 버퍼에 저장하여 질의 처리를 빠르게 하기 위한 모듈이다. <그림 7>은 공간 데이터를 저장하는 데이터 버퍼의 자료 구조를 보여준다.

```
typedef struct
{
    double x; //double x
    double y; //double y
} Vertex;
typedef struct
{
    int nVertex; //Vertex (x,y)의 개수
    Vertex *pVertex; //Vertex Data
    CString Label; //Label
    int *pMultInfo; //멀티폴리곤, 멀티폴리라인 정보
} Vertexs;
typedef struct DBN
{
    int DBuffID; //데이터 버퍼 아이디
    int UpdateFlag; //-1 (버퍼 생성중) 1 (사용가능 상태)
    int nRecord; //레코드 수
    int *fid; //FID 저장객체
    Vertexs *pVertexs; //공간 데이터 저장객체
    struct DBN *next; //다음 DataBufferNode 포인터
} DataBufferNode;
```

<그림 7> 데이터 버퍼의 자료 구조

<그림 7>의 자료 구조에서 구조체 DBN 내의 DBuffID는 메타 데이터와 연동하기 위한 데이터 버퍼 ID이고, UpdateFlag는 데이터 버퍼 생성 중에 질의 처리자에 의한 접근을 방지한다. 데이터 버퍼에 기록되는 총 레코드 수는 nRecord 변수에 기록되고, 그 수만큼 Fid 배열과 Vertexs 구조체를 생성한다. <그림 8>은 데이터 버퍼 생성 다이얼로그에서 데이터 버퍼를 생성하는 것을 보여준다.



<그림 8> 데이터 버퍼 생성 다이얼로그

4.4 인덱스 관리자

인덱스 관리자는 XServer의 내부 인덱스를 이용해서 후보 객체를 검색한다. 내부 인덱스의 생성/갱신/관리와 관련된 인덱스 관리자의 주요 함수는 <표 2>와 같다.

<표 2> 인덱스 관리자의 주요 함수

함수	내용
int RtreeRealloc()	R-tree 구조체의 메모리 할당
int RowidRealloc(RtreeInfo *RTI, int n)	Rowid 구조체의 메모리 할당
int RtreeFullLoad(MetaManager *MMG)	XServer 시작시 레이어(레이어)의 R-tree를 초기화 / 생성
int RtreeLoadPolyline(int nRecord, string tablename)	공간 타입을 R-tree 로드 함수
int RtreeLoadPoint(int nRecord, string tablename)	공간 타입을 R-tree 로드 함수
int insertRectChar *rowid, double *MBR, int RID)	insert 질의 처리후 변형사항을 해당 R-tree에 반영
int insertRectChar *rowid, double X, double Y, int RID)	insert 질의 처리후 변형사항을 해당 R-tree에 반영
int UpdateRectChar *rowid, double X, double Y, int RID, int NodeId)	Update 질의 처리후 변형사항을 해당 R-tree에 반영
int UpdateRectChar *rowid, double *MBR, int RID, int NodeId)	Update 질의 처리후 변형사항을 해당 R-tree에 반영
int DeleteRect(CString Rowid, int RID, double *MBR)	Delete 질의 처리후 변형사항을 해당 R-tree에 반영

인덱스 관리자에서 사용되는 내부 인덱스의 노드는 MBR 정보와 노드 포인터로 구성된다. 리프 노드의 노드 포인터는 Oracle에 저장된 객체의 Rowid를 지정한다. 인덱스 관리자의 공간 검색 질의와 관련된 주요 함수는 <표 3>과 같다.

<표 3> 인덱스 관리자 검색 관련 주요 함수

함수	내용
void SetRect(RtreeInfo *RtreeF, int RID)	검색 대상이 되는 R-tree set
int SearchRtreeContain(double xmin, double ymin, double xmax, double ymax)	MBR Contain 공간 연산
int SearchRtreeOverlap(double xmin, double ymin, double xmax, double ymax)	MBR Overlap 공간 연산
int SearchRtreeDisjoint(double xmin, double ymin, double xmax, double ymax)	MBR Disjoint 공간 연산
int SearchRtreeIntersectObject(double xmin, double ymin, double xmax, double ymax, double bxmin, double bymin, double bxmax, double bymax)	내부 MBR 공간 연산을 추가한 Intersect 공간 연산
int SearchRtreeContainObject(double xmin, double ymin, double xmax, double ymax, double bxmin, double bymin, double bxmax, double bymax)	내부 MBR 공간 연산을 추가한 Contain 공간 연산
int SearchRtreeTouchObject(double xmin, double ymin, double xmax, double ymax, double bxmin, double bymin, double bxmax, double bymax)	내부 MBR 공간 연산을 추가한 Touch 공간 연산
int SearchRtreeOverlapObject(double xmin, double ymin, double xmax, double ymax, double bxmin, double bymin, double bxmax, double bymax)	내부 MBR 공간 연산을 추가한 Overlap 공간 연산
int SearchRtreeDisjointObject(double xmin, double ymin, double xmax, double ymax, double bxmin, double bymin, double bxmax, double bymax)	내부 MBR 공간 연산을 추가한 Disjoint 공간 연산

4.5 DB 관리자

DB 관리자는 OCCI를 통해 Oracle에 공간 데이터를 저장하고 검색한다. Oracle에 공간 데이터를 저장하기 위해서 점은 Oracle의 기본 타입을 활용하고, 폴리라인과 폴리곤은 사용자 정의 타입을 정의하여 활용한다. <그림 9>는 폴리라인과 폴리곤을 저장하기 위해 사용한 사용자 정의 타입을 생성하는 모습이며, <그림 10>은 사용자 정의 타입의 컬럼을 가지는 테이블 생성하는 모습이다.

```
CREATE TYPE XNUMBERS AS VARRAY(2000) OF NUMBER;
CREATE TYPE XPOINTS AS OBJECT (NUMBERS XNUMBERS);
```

<그림 9> 사용자 정의 타입 생성

```
CREATE TABLE POLYLINE_TABLE
(FID NUMBER, COUNT NUMBER,
GEODATA XPOINTS, LABEL VARCHAR2(30));
```

<그림 10> 공간 테이블 생성

<그림 10>에서 FID는 객체 ID를 저장하고 COUNT는 공간 데이터를 구성하는 점의 개수를 저장한다. GEODATA는 공간 데이터를 저장하고, LABEL은 각 객체의 이름을 저장한다.

OCCI를 통해 사용자 정의 타입에 접근하기 위해서는 OTT 프로그램을 이용해 C++의 클래스로 변환해 주어야 한다. 그림 11은 사용자 정의 타입을 C++ 클래스 화일로 변환하기 위한 설정 화일이며, <그림 12>는 C++ 클래스 화일을 생성하기 위한 OTT 명령문이다.

```
CASE=SAME MAPFILE=OraOciTypemap.cpp
TYPE XNUMBERS HFILE=OraOciType.h cppfile=OraOciType.cpp
TYPE XPOINTS HFILE=OraOciType.h cppfile=OraOciType.cpp
```

<그림 11> INTYPE File (OraOciTypein.typ)

```
ott userid=scott/tiger intype=OraOciTypein.typ
outtype=OraOciTypeout.typ code=cpp attraccess=private
```

<그림 12> OTT 명령문

<그림 13>은 Oracle에 접근해서 사용자 정의 타입의 객체를 가져오는 예제이다.

```
// 오라클을 사용할 수 있도록 Environment 생성
env = Environment::createEnvironment (Environment::OBJECT);
OraOciTypemap(erv); //OTT로 생성한 class 적용
con = env->createConnection(user, passwd); //Oracle 접속

Statement stmt = con->createStatement(stmtSQL); //Statement 생성
ResultSet resultSet = stmt->executeQuery(); //질의 실행, 결과 포인터 가져옴
while(resultSet->next()) //질의 마지막 결과까지 Loop 돌면서 데이터 가져옴
{
    XPOINTS *o = (XPOINTS *)ODB->resultSet->getObject(7);
    const vector<Number> &points = o->getNumbers();
}
stmt->closeResultSet(resultSet); //결과 포인터 닫음
```

<그림 13> OCCI 검색 질의 코드 예제

<그림 13>에서 OraOciTypemap 함수는 OTT로 변환한 클래스를 OCCI 환경에서 사용 가능하게 한다. executeQuery 함수는 SQL의 SELECT문을 실행하는 함수이다. 결과 중 사용자 정의 타입 객체를 가져오기 위해서 getObject 함수를 사용한다.

4.6 메타 데이터 관리자

메타 데이터 관리자는 질의 처리시 인덱스

사용 여부나 데이터 버퍼 사용 여부 등과 같이 질의 처리 관리자에서 참조하는 모든 정보를 제공하는 모듈로서 Oracle의 특정 테이블에 저장된 메타 데이터 정보를 XServer 기동시 가져오게 된다. 메타 데이터는 질의 처리 관리자에서 입력/갱신/삭제의 질의가 일어나면 갱신된다. 메타 데이터의 구조는 <그림 14>과 같다

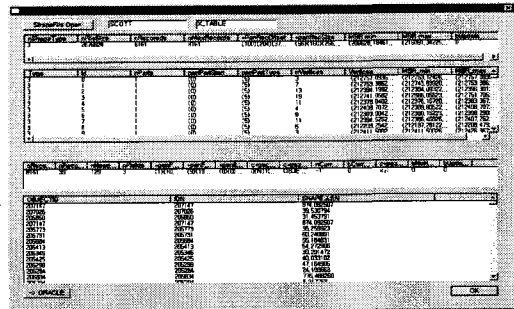
```
typedef struct MT
{
    CString *TableName; //레이어 이름
    int TableType; //공간 속성 타입
    double MBR[4]; //레이어 전체 MBR
    int nRecord; //레코드 총 갯수
    int RtreeID; //외부 RtreeID (-1: 해당 테이블 외부 Rtree 없음)
    int DBuffID; //데이터 버퍼 ID (-1: 해당 테이블 데이터 버퍼 없음)
    int nAtt; //속성 갯수
    CString *AttName; //속성 이름
    char *AttType; //속성 타입
}MetaTableNode;

typedef struct MS
{
    CString *SchemaName; //스키마 이름
    MetaTableNode *pMetaTB; //레이어 목록 포인터
    struct ML *next;
}MetaSchemaNode;
```

<그림 14 > 메타 데이터의 구조

4.7 수입/수출 관리자

수입/수출 관리자는 Shape 화일에서 데이터를 추출하여 Oracle에 저장하거나 Oracle에서 데이터를 추출하여 Shape 화일로 저장하기 위한 모듈로서 <그림 15>는 수입/수출을 담당하는 수입/수출 관리자 다이얼로그이다.



<그림 15> 수입/수출 관리자 다이얼로그

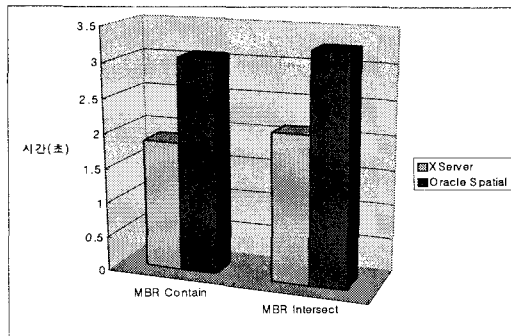
Shape 화일의 수입시 Shape 화일의 헤더의 공간 및 속성 데이터 타입 정보를 활용하여, 메타 데이터 관리자의 메타 데이터에 스키마 이름, 레이어 이름, 공간 데이터 타입, 전체 테이블의 MBR, 공간 데이터의 총 개수 등의 정보를 저장한다. 그리고, DB 관리자를 통해 해당 공간 테이블과 속성 테이블을 생성하고 데이터를 기록한다.

5. 성능 평가

본 장에서는 XServer 시스템과 Oracle Spatial과의 성능을 비교하였다. XServer와 Oracle Spatial에서 성능 비교를 위해 사용한 연산은 MBR Intersect와 MBR Contain이며, 각각을 단일 레이어와 복합 레이어에서 수행한 후 결과를 가져오는 시간을 측정하였다.

5.1 단일 레이어 검색 성능 평가

단일 레이어에 저장된 공간 객체 타입은 폴리라인이고 수는 8161개이다. <그림 16>은 단일 레이어에서의 MBR 연산에 따른 검색 질의 성능을 비교한 그래프이다.

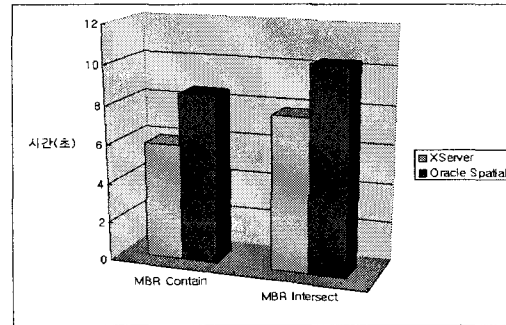


<그림 16> 단일 레이어 MBR 연산 성능 비교

실행 결과를 통해 MBR Contain 연산에서 XServer가 Oracle Spatial 보다 평균 1.65배의 성능 향상을 보임을 알 수 있다. 또한, MBR Intersect 연산에서 XServer가 Oracle Spatial 보다 평균 1.54배의 성능 향상을 보임을 알 수 있다.

5.2 복합 레이어 검색 성능 평가

복합 레이어에 저장된 공간 객체 타입은 폴리라인과 폴리곤이고 수는 44220개이다. <그림 17>은 복합 레이어에서의 MBR 연산에 따른 검색 질의 성능을 비교한 그래프이다.



<그림 17> 복합 레이어 MBR 연산 성능 비교

실험 결과를 통해 MBR Contain 연산에서 XServer가 Oracle Spatial 보다 평균 1.44배의 성능 향상을 보임을 알 수 있다. 또한, MBR Intersect 연산에서 XServer가 Oracle Spatial 보다 평균 1.33배의 성능 향상을 보임을 알 수 있다.

6. 결론

본 논문에서는 대용량 지도 서비스를 보다 효율적으로 서비스할 수 있는 Oracle 기반의 XServer 시스템을 설계 및 구현하였다.

XServer는 질의 처리 관리자로 다양한 클라이언트의 질의를 효율적으로 처리하고, 인덱스 관리자의 내부 인덱스를 이용한 MBR 연산과 데이터 버퍼 관리자에 의해 대용량의 공간 데이터를 빠르게 검색한다. 수입/수출 관리자는 대용량의 공간 데이터를 추출하고, DB 관리자는 XServer와 Oracle간 효율적인 질의 처리를 하게 한다.

본 논문에서는 XServer의 성능 평가를 위해서 지리 정보를 저장 및 검색할 수 있는 상용 데이터베이스인 Oracle Spatial과 비교 실험을 수행하였다. 실험 결과를 통해 XServer가 Oracle Spatial 보다 MBR 검색에서 약 1.4배의 성능 향상을 보였다.

< 참고 문헌 >

- [1] E. Clementini, J. Sharma, M. Egenhofer, "Modeling Topological spatial Relations: Strategies For Query Processing," Int. J. of Computer and Graphics, 1994, VOL.18, NO.6, pp.815-822.
- [2] D. Geringer, Oracle 9i: Spatial, Oracle, 2001.
- [3] M. F. Goodchild, "Finding the Mainstream," ScanGIS, 2003.
- [4] R. K. Kothuri, S. Ravada, "Efficient Processing of Large Spatial Queries Using Interior Approximations," Proc., The 7th International Symposium on Advances in Spatial and Temporal Databases, 2001, VOL.2121, pp. 404-421.
- [5] C. Murray, Oracle Spatial User's Guide and Reference, Release 9.2, Oracle, 2002.
- [6] D. Raphaely, J. Gregoire, Oracle C++ Call Interface, Oracle, 2001.
- [7] C. Shahabi, "Introduction to Spatial Database Systems," Ralf Hart Hartmut Guting's VLDB Journal, 1994, VOL.3, NO.4.
- [8] Autodesk GIS Design Server White Paper: Optimizing Oracle Spatial, <http://www.autodesk.com/gisdesignserver>, Autodesk Inc, 2002.
- [9] 윤재관, 장염승, 한기준, "모바일 GIS를 위한 위치 기반 서비스," 한국정보과학회지 데이터베이스 연구지, 2002, 18권1호, pp.3-15.
- [10] 이상원, "멀티미디어 데이터를 위한 확장형 인덱스 소개," 데이터베이스연구학회지, 2003, 19권3호, pp.1-9.