# GML Database Based On Oracle XML DB

Lan Xiaoji, Ren Jintong
School of Architectural and Surveying& Mapping Engineering
Jiangxi University of Science and Technology
Ganzhou, China
landcom8835@163.com jintongabc@163.com

Cai Jinbao
School of Foreign Language and International Trade
Jiangxi University of Science and Technology
Ganzhou, China
Caijinbao1971@163.com

*Abstract*-**Based on the study of XML documents storage in Oracle XML DB, this paper conducts a research on extending Oracle XML DB to store and manage GML documents. Meanwhile, owing to the defects that the XML DB cannot support the spatial index, an external R-Tree index is added to Oracle database to improve the efficiency of GML spatial query and operation. At the same time, with the help of NTS as well as with Oracle's own support XQuery, the spatial extension to XQuery in Oracle is proposed and implemented. Through the research of this paper, we provide a solution of GML database for GML storage, index and query.**

***Keywords- Oracle XML DB; GML database; GML storage; GML query; XQuery extension***

## I. INTRODUCTION

GML (geography markup language) has been widely focused on by the GIS experts, scholars and developers since it was proposed by the OGC (Open Geospatial Consortium). GML is a Markup Language used to describe the geographical objects in practice. It provides a perfect solution for spatial data sharing and interoperability. In recent years，Web data has massively emerged on the Internet, especially. Due to the great development of Web Service, GML has been widely used for the exchange and transmission in the field of geospatial data which has played multi-roles in the Web services. In addition, a series of standard geospatial Web services such as Web Map Service (WMS), Web Feature Service (WFS) and Web Coverage Service (WCS), have been developed by OGC. GML can be used for the messages request and response in WFS.

GML has become the de facto international standards of the spatial data encoding, transmission, storage and distribution. With the increasing application of GML, a large number of GML data are constantly emerged. Therefore, how to effectively use and manage the immense data is a problem needed to be resolved urgently in the field of GIS. This paper intends to use the Oracle XML DB technology of Oracle Corporation to effectively store and manage GML spatial data.

## II. CURRENT STATUS OF GML RESEARCH

Many experts and scholars have specialized in the research of GML due to its advantages. Researcher and developer have mainly focused on the GML spatial data model, data exchange, spatial data visualization, GML-based spatial information integration and sharing and so on. Furthermore, GML storage and management have been researched with the increase of GML data.

Guan Jiehong, Li Jun [1] intended to use XML enabled database to manage the GML spatial data, and Yu li [2] presented the approach of storing GML spatial data by key element into an object-oriented database. Lan Xiaoji [3-5] took the GML feature as the storage granularity, used the native xml database-Ozone for the GML feature storage, and provided the idea of extending preorder-postorder GML document encoding; and combined with the traditional spatial data index, an integrated index mechanism based on spatial index and R+ Tree has been presented. In the terms of the GML spatial data query, based on the XQuery, he has made an extension to it with spatial data type and spatial operation functions. The national 863 project undertaken by Wuhan University and Fudan University conducted a research on GML storage and spatial data integration. Corcoles [6] has compared the efficiency of GML storage and query (SQL) which stored in the three relational databases. Based on SQL, a GML spatial query language was suggested [7]. Due to the huge differences between XML data model and traditional relational model, it is not the most effective way to extend the SQL for supporting GML query because it is very time-consuming when users have to make some conversions between GML and the relational database and would lower the processing speed of the database. In addition, this processing is incompatible with the XML query language. Besides, Vatsavai [8] has made a comparison among some XML query languages and advocated to extend XQuery. However, this paper did not aim at addressing how to resolve the problem. It is just a hypothesis.

The research on GML spatial database is still in a preliminary stage. More intensive and specific work is needed to continue. This paper aims at storing and managing GML spatial data by the use of Oracle XML DB and intends to extend Oracle XQuery to make it suitable for the GML spatial data through NTS technology and the support of XQuery. By the way of establishing index of R-Tree on the exterior of Oracle, the operating efficiency of GML spatial query will be greatly enhanced. Taken the unique advantage of Oracle database which is powerful in the transaction processing and controlling, it provides an efficient solution to establish a complete, efficient and large-scaled GML spatial database.

This research was funded by the National Natural Science Foundation of China under Grant No. 40971234, 40761017，40401045, and the Science and Technology Project of Jiangxi Education Department under Grant No. 195 of Ganjiaojizi [2006], and the Open Foundation of Jiangsu Provincial Key Lab of Geographical Information Sciences under Grant No. JK20050302.

978-1-4244-4994-1/09/$25.00 ©2009 IEEE

## III. GML STORAGE AND MANAGEMENT BASED ON ORACLE XML DB

### A. *A brief introduction to Oracle XML DB*

From the second edition of Oracle 9i database, Oracle XML DB has been seamlessly integrated with the Oracle database. In order to support XML, Oracle XML DB introduces the object type of XMLType, which is a native server data type. Database can be understood as a column or a table containing the XML, just as the data type of Date. Therefore, we can use the XMLType table or column to store XML data. The use of such native XMLType data type to store XML data can have the same effect as the Native XML Database.

The research in this paper is based on the latest version of Oracle--Oracle 11g. In Oracle 11g, Oracle XML DB has achieved another leap forward, such as a new binary XML storage model, XMLIndex binary XML and unstructured XML storage model index. In addition, Oracle 11g also provides an XQuery engine to enable the Application developers use the preferred API (such as JDBC, ODP.NET, and Web services) to make full use of Oracle database XQuery function [9].

The abstract data type of XMLType in Oracle XML DB, provides different storage models for us. XMLType tables and columns can choose the following storage methods: Structured storage, unstructured storage and Binary memory storage [9].

### B. *XML Storage and Management in Oracle XML DB*

Oracle XML DB supports three XML storages. Which kind of storage is suitable for your solution needs to be considered carefully. This paper tests the storage and query efficiency in order to provide some evidence to resolve this problem. The experiments are performed on Intel (R) Core (TM) 2 Duo T6400@2.00GHz system running Windows Vista Home Basic with 2.00 GB of main memory.

#### *(1) Storage efficiency*

Using the following PL / SQL statement, the test is performed to store the XML data:

*Insert into player*

*Values (xmltype(bfilename('xmldir','Player0.5.xml'), nls_charset_id('AL32UTF8')));*

The results of storage efficiency are shown in Table I. It is can be seen from Table I that, with the increasing of XML document size, the required storage time has also increased; As for the three storage methods, the Structure Storage is the fastest when storing the same size XML document and the efficiency of Binary Memory Storage is higher than the Unstructured Storage. The XML document has to be analyzed into a binary mode when storing a XML document. Then it is stored into the XML DB. For that XML storage which has the stringent requirements on the XML format and mode, it is advisable to use the Structure Storage. Besides, as for the storage of an entire XML document or a partial one, it is available to use the binary storage mode. Moreover, the Unstructured CLOB Storage is not recommended to put into use.

TABLE I. COMPARISON OF STORAGE EFFICIENCY

| Document Size(MB) | THE TIME OF STRUCTURED STORAGE(S) | THE TIME OF UNSTRUCTURED STORAGE(S) | THE TIME OF BINARY STORAGE(S) |
|---|---|---|---|
| 0.5 | 0.190 | 0.522 | 0.230 |
| 1.0 | 0.233 | 0.755 | 0.390 |
| 1.5 | 0.380 | 1.220 | 0.491 |
| 2.0 | 0.518 | 1.831 | 0.712 |
| 2.5 | 0.577 | 2.783 | 0.843 |
| 5.0 | 1.298 | 4.158 | 1.842 |
| 10.0 | 2.933 | 7.350 | 4.013 |

#### *(2) Query efficiency*

The results of query efficiency are shown in Table II. It is can be seen from Table Ⅱ that, with the increasing of XML document size, the query time required has also increased; To query the same size XML document, the structure storage is the fastest, binary memory storage efficiency is higher than the unstructured storage. For that XML storage which has the stringent requirements, it is advisable to use the Structure Storage which is based on XML Schema. If XML documents have no fixed schema, you can choose to use binary storage mode.

TABLE II. COMPARISON OF QUERY EFFICIENCY

| Document Size(MB) | TIME OF STRUCTURED STORAGE(S) | TIME OF UNSTRUCTURED STORAGE(S) | TIME OF BINARY STORAGE(S) |
|---|---|---|---|
| 0.5 | 0.027 | 0.384 | 0.112 |
| 1.0 | 0.033 | 0.819 | 0.211 |
| 1.5 | 0.044 | 1.031 | 0.274 |
| 2.0 | 0.059 | 1.853 | 0.353 |
| 2.5 | 0.078 | 2.082 | 0.421 |
| 5.0 | 0.241 | 5.082 | 1.372 |
| 10.0 | 0.498 | 9.488 | 2.123 |

### C. *GML storage and management in Oracle XML DB*

Due to the fact that GML is based on the XML encoding, through the study of GML storage and management by the Oracle XML DB, different storage methods can be selected to store different GML data. GML document has a good structure ensured by GML Schema, which defines the structure and meta-data for the GML document. Therefore, the GML storage and management consists of two parts: the storage of GML Schemas and the storage of GML documents.

#### *(1) GML Schema storage and management*

GML is not a single schema. It is composed of GML core schemas, GML application schemas and GML instance documents. GML core schema is defined by OGC while GML application schema is defined by GML core schemas. The data producers provide GML instance documents, and its content and structure must conform to GML application schema.

As for the storage and management of GML schemas, the PL/SQL statement DBMS_XMLSCHEMA.Register_Schema() can be called to register GML schemas. After the GML schema registration, a default table (an XMLType table) for each global element will be created, that is, the object table. Each row in the table is expressed by an XMLType instance.

In Oracle XML DB, an object model can be derived from information of GML schemas; the object model can be used to decompose GML document into a series of objects, and then are stored in the database. Oracle XML DB does not set all GML schemas as the default schemas of database system; as a result, in practical use of XML DB storage about the GML data, it is necessary to register GML core schemas and GML application schemas, in order to store GML instance documents by structured way.

### *(2) GML documents storage and management*

For the reason that GML documents should conform to their own GML application schemas, and through the above tests, it is more reasonable to choose the structured storage for GML storage. At the same time, to create suitable GML spatial data storage model according to GML documents, and to divide it according to the storage granularity size, to select the appropriate granularity size are conducive to make a query and the corresponding operation on GML spatial data.

Granularity size is the polymerization degree of data summarized, which determines not only the size of the fragments of each GML document broken down in the database but also directly determines GML processing speed and efficiency. According to the granularity size, there are three ways to store GML into Oracle database: (1) coarse-grained method; (2) middle-grained method; (3) fine-grained method. Among them, the second method is adopted in this paper. In the GML schema, each single feature type is a complete description of a certain type of the geographic entities. Therefore, we choose middle-grained method to store GML, which storage granularity is one feature.

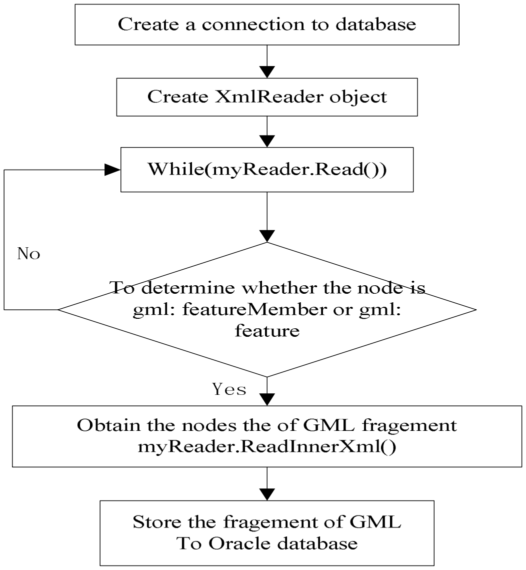
Create a connection to database
Create XmlReader object
While(myReader.Read())
No
To determine whether the node is gml: featureMember or gml: feature
Yes
Obtain the nodes the of GML fragement myReader.ReadInnerXml()
Store the fragement of GML To Oracle database

Figure 1. The process of GML feature parsing and storage

Another important issue needs to be resolved during the procedure of GML storage is the GML parsing, which is the basis of other operations. Through the GML document parsing, the feature-based GML fragment can be used to perform storage and query on the GML data. The GML parsing in this paper is based on the .NET streaming model, which is read-only, forward-only and no buffering. It uses XmlReader to parse GML documents. The feature-based GML parsing and storing procedure is shown in Fig. 1.

## IV. GML SPATIAL INDEX

The technique and method of spatial index is one of the key technologies of spatial database. Many scholars have carried out extensive studies on the spatial data index, and have provided a large number of index structures like the grid file, quad tree series, R-Tree series and so on. The research of this paper is conducted on the basis of the R-tree spatial index. Two methods are available to carry out the spatial index. The first one is the database-internal spatial index which is the part of the database system. The second one is the database-external spatial index and managed separately outside the database system. Combining the existing B-tree index of Oracle over the XML and the flexibility of spatial index on migration and maintenance, the second approach has been used in this paper.

It is not necessary that each feature has to possess an ID and MBR in the GML specification. But in building R-tree index, the feature's ID (or FID) and MBR are demanded. The procedure of creating R-tree index is shown in Fig. 2. The efficiency of spatial query will be greatly improved by the spatial index technology. When querying GML data, FID should be found by R-tree index first, and then the related data of FID are extracted from the GML database.

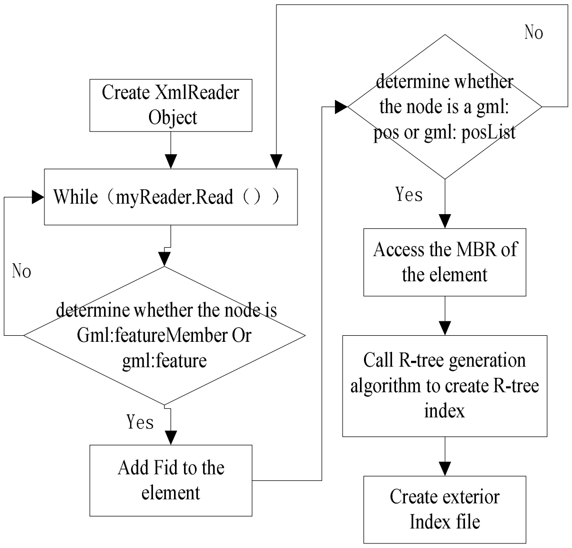
Create XmlReader Object
While（myReader.Read（））
No
determine whether the node is Gml:featureMember Or gml:feature
Yes
Add Fid to the element
No
determine whether the node is a gml: pos or gml: posList
Yes
Access the MBR of the element
Call R-tree generation algorithm to create R-tree index
Create exterior Index file

Figure 2. The process of creating R-Tree index

## V. SPATIAL EXTENSION TO XQUERY

XQuery is an XML query language specification which is proposed by W3C. Oracle Database 10g Release2 debuts a full-featured and high performance database-native XQuery engine. The JTS Topology Suite is an API for modeling and manipulating 2-dimensional linear geometry. It provides numerous geometric predicates and functions. NTS (.Net Topology Suite) can be considered as the .NET version of JTS. Because the Oracle XQuery is only suitable for the simple XML, not for the GML spatial query, that is to say it is lack of spatial data types and spatial operations, so the NTS has been used as the spatial operations library to extend XQuery in .Net environment. The extended XQuery follows its grammatical structure completely and it has just added some spatial data types and spatial operations. The XmlQuery () and XmlTable () which provided by Oracle are used to perform the XQuery operation. The following example is the GML fragment with the area of more than 75,000 by using the Oracle XQuery query:

```
select xmlquery ('declare namespace fme="http://www.safe.com/gml/fme"
;
for       $i in/fme:States
where $i/fme:AREA >75000
return /*'
passing object_value  returning content)
from gmltest;
```

This Query presents the attribute query without adding some spatial operator while the following one is a spatial query with adding the spatial operator:

```
select xtab.column_value
from gmlcity, xmltable (
        'declare namespace gml="http://www.opengis.net/gml";
declare namespace fme="http://www.safe.com/gml/fme";
for  $i in/fme:Cities
where contains（$i,” 21.3744664370352 -157.92001385484”）
return /*'
passing object_value ) xtab;
```

To the XQuery language with spatial operators added, the first step is to use the analytic engine to parse the query language. When it encounters one spatial operation function, it calls for the appropriate handler for processing the query. As for the above query, the process is illustrated in Fig. 3.

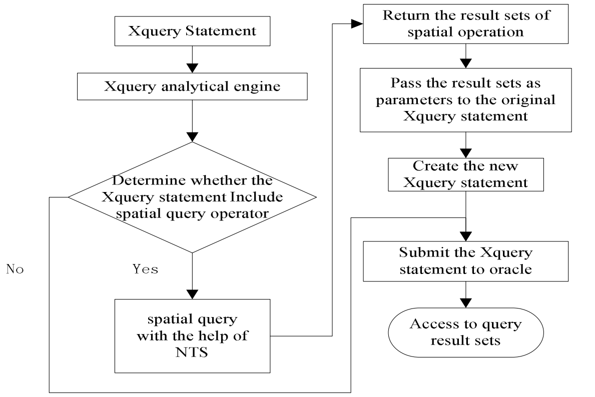
Xquery Statement
Xquery analytical engine
Determine whether the Xquery statement Include spatial query operator
No
Yes
spatial query with the help of NTS
Return the result sets of spatial operation
Pass the result sets as parameters to the original Xquery statement
Create the new Xquery statement
Submit the Xquery statement to oracle
Access to query result sets

Figure 3. The process of spatial query of GML XQuery

## VI. CONCLUSIONS

On analyzing Oracle XML storage and management, the extensions to Oracle XML DB has made it available for the storage and management of GML spatial data, and an index method which is file-based outside the Oracle database has been carried out via R-Tree index. In this paper, the efficiency of GML spatial query has been presented. Due to the fact that the XQuery language is accepted by Oracle, the XQuery query language has been extended and some spatial operators are added to it based on NTS.

## REFERENCES

[1] Li Jun, Guan Jiehong and Li Yuzhen, “G2SDB: A new storage model for GML documents”, Geomatics and Information Science of Wuhan University, vol. 29, no. 12, pp. 1071-1074, Dec 2004. (in Chinese)

[2] Yu Li, Bao Peiming and Zhang Shuliang, “Research in Storage Management of GML Spatial Data”, Journal of Nanjing Normal University (Engineering and Technology Edition) , vol. 6, no. 1, pp. 67-71, Mar 2006. ( In Chinese)

[3] Lan xiaoji, Zhang Shuliang, Liu Deer and Lu Guonian, “Study on GML spatial database”, Science of Surveying and Mapping, vol. 30, no. 5, pp. 16-19, Oct 2005. ( In Chinese)

[4] Lan Xiaoji, Lu Guonian and Liu Deer, “Research on Query and Index for GML Spatial Data”, Journal of Remote Sensing, vol. 10, no. 6, pp. 854-863, Nov 2006. ( In Chinese)

[5] Lan xiaoji, Liu Deer, Zhang Shuliang and Jiang Yongfa, “Study on GML Spatial Data Querying System”, Geotechnical Investigation & Surveying, vol. 4, pp. 52-55, May 2005. ( In Chinese )

[6] J. E. Córcoles and P. González, “Analysis of different approaches for storing GML documents”, Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, Mclean, Virginia, USA, pp. 11-16, Eds. New York: ACM, NY, USA, 2002.

[7] J. E. Córcoles and P. González, “A specification of a spatial query language over GML”, Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems, Atlanta, Georgia, USA, pp. 112-117, Eds. New York: ACM, NY, USA, 2001.

[8] Ranga Raju Vatasvai, “GML-QL: A Spatial Query Language Specification for GML”, unpublished. http://www.cobblestoneconcepts.com/ucgis2summer2002/vatsavai/vatsavai.htm, 2002.

[9] Drew Adams, Oracle XML DB Developer's Guide, 11g Release 1, Oracle Cor., May 2008 (http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28369.pdf)