ISSN No. 0976-5697

Volume 2, No. 5, Sept-Oct 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Conceptual Review on Relational and Spatial Database Query Processing and Benchmarking

Ravi Kant Vyas*, Manoj Paliwal and B.L. Pal
Mewar University
Chittorgarh (Raj) India
vyasravikant@gmail.com

***Abstract*:** Applications of different areas need a deal with databases and quantitative and qualitative temporal constraints. Issues of relational database are re-presentation of temporal data, grouping identifiers, primary temporal relations keys and temporal integrity constraints. Purpose of the paper is to investigate the characteristics that distinguish spatial databases systems from traditional ones. An efficient knowledge representation for classification of spatial data is based on both non-spatial properties of the classified objects and attributes, predicates and functions describing spatial relations between classified objects and other features located in the spatial proximity of the classified objects. The great popularity of geospatial data and increasing use in commercial as well as scientific applications is a significant event of our times. The flow of data has made DBMS a ubiquitous component for research. Although many DBMS have been built, there have been very less systematic comparisons of functionalities and performance especially few benchmark standards.

Keywords: database, relational, spatial, benchmarking

I. INTRODUCTION

Many applications in different areas need to deal with both database and temporal constraints. Dealing with temporal constraints arises from areas, ranging from office information systems to artificial intelligence fields such as scheduling diagnosis, planning, and knowledge representation.

An approach that integrates the advantages of DBMS and AI temporal managers, we choose the relational model to deal with data as:

a. Relational databases are standard to deal with data.
b. there is a very strong formalization and theoretical background, and
c. Good deal of research to extend relational model to deal also with temporal information.

A temporal database have a maintain time varying and time dimension, data in contrast to conventional database which carries only the current data. There is a growing body of research in querying temporal data and modeling data.

We focus on the relational data model and comparing database states at two different time points, capturing the periods for concurrent events and accessing to times beyond these periods, handling multi-valued attributes, temporal grouping and coalescing and restructuring temporal data, etc.

Data protection is ensured by components of a (DBMS). An access control mechanism ensures data confidentiality. To access a data object, the rights of the user against a set of authorizations is checked, done some security administrator. When a subject tries to modify some data, mechanism verifies the user right to modify the data.

With the need for people to store complex data such as multi-media data, the relational database had to be extended with support for customized data types, functions and indexes these were the object relational databases or ORDBMS. An extension to the SQL standard, SQL-99, needed to reflect new capabilities and allow the user to utilize them in a declarative manor.

In case of Spatial Database system should it should contain an elegant framework, s general as possible and not designed for one particular area of applications and also have a formally defined semantics that is closed under set theoretic, geometric and topological operations and that is defined in terms of finite representations; also it is independent of a particular (DBMS) but co-operative with any DBMS. It should have efficient implementation techniques, especially for the operations on n-dimensional objects and have an up-to-date visual user interface and a gateway to multimedia.

A little attention is given to the issues of spatial data. This is due to the lack of interest or could be the vast know-how that is needed in three totally different areas of science i.e. databases and information systems, geography and cartography, abstract and computational geometry and topology.

Geographic data consists of non-spatial objects and spatial descriptions of these objects. Non-spatial descriptions of spatial objects can be stored in a relational database where one attribute is a pointer to the spatial description of the object, as in.

Spatial relation is extensively used in a spatial database query language, content-based spatial data retrieval and spatial data analysis, and it is an important factor for spatial data mining.

II. OBJECT RELATIONAL QUERY PROCESSING

There are several features of an ORDBMS that has to be properly handled by the object-relational query optimizer and illustrate how they should be supported by the object relational optimizer using the examples from. An example from will also be introduced. example in the text will be the table *emp* created to hold instances of the row type *employee_t* as shown below:
create row type employee_t(id employee_id, name varchar(30), salary int, startdate date, location point, picture image);

A. Create Table Emp of Type Employee_t:

The user-defined type point above is created to model points in 2-D in a more simple way than adding the two attributes longitude and latitude as floating point numbers.

a) User-defined operators and selectivity functions:

As opposed to relational databases where the selectivity functions for built-in operators is hard-coded into the system this knowledge has to be explicitly defined for user defined operators in ORDBMS for the optimizer to do a good job.

For example, given the query below containing a call to the user -defined operator, N_equator_equals, the user must also specify a corresponding selectivity function for the function:

B. Select Name from Emp Where Location N_Equator_Equals Point ('200,2');

In the ORDBMS there should be functionality to associate this user-defined selectivity function to the operator. During optimization time when the optimizer encounters the call to *N_equator_equals* it will call the correct selectivity function and use the returned value (floating-point number between 0-1) to get a measurement of the cardinality of the user-defined function. Often the selectivity function use statistics from the ORDBMS to calculate its value.

a) User-Defined Access Methods:

An access method is a collection of functions for handling indexes on data such as open a scan of an index, iterate through the scan, insert, delete or replace a record and close the scan. Consider the query below extracting the employees that lives within the area defined as the bounding box specified by origin and point(2,1):

C. select name from emp where location contained box('0,0,2,1'):

Since finding all the point contained within the bounding box is a 2-dimensional search there is a need for a 2-dimensional index such as e.g. an R-tree.

A plain B-tree will not do. Since an ORDBMS allows the creation of user-defined types, e.g. point, special access methods are needed to speed up access to data defined in terms of these types. However, different data types using the same index in their access methods can assign different semantics to the operators (<,>,= e.t.c) of the particular index.

Also, the access method must handle tasks like locking on index objects, recovery of index data structures and coordination with the ORDBMS buffer manager when reading and writing index disk pages.

III. SPATIAL CLASSIFICATION KNOWLEDGE BASE DESIGN

a) Spatial Predicates:

Spatial predicates are constraints used to describe the relationship between spatial objects in the form of R(x, y), that is the dual function of returning a value of true and false. Position, direction and distance are important content to the concept of spatial metadata, and are the most important component of description, expression and access to GIS position association rules.

According to their statements GIS spatial predicate is divided into three categories:

a. Topology, such as to-disjoint, to-intersect, to-contain, to-inside, to-meet, to-equal, to-cover and to-cover-by.
b. Direction, such as direct, east-of, west-of, south-of, north-of, southeast-of, northeast-of, southwest-of and northwest-of.
c. Distance, such as dist, close-to, near-to and far-away.

b) Table Structure Design:

Rules are composed of premises and conclusions where the Premises are the assumptions posed by a series of connected facts, and conclusions are tag information for rules of category. In order to avoid the ambiguity of interpretation of the rules, the premises of rules can only be constituted by "and" operator.

a. Rule Table: record the information of rules such as degree of support, confidence level.
b. Fact Table: record the fact information that rules can no longer divided.
c. Character Table: record the latter part of the eigenvalues or spatial coordinates connected by the predicates.
d. Predicate Table: record all the predicate information of rules.
e. Conclusion Table: record the conclusions of the rules.
f. Rule Index Table: record the index information that the rules of the fact contains.

IV. BENCHMARK MECHANISM

Benchmark is based on the following criteria:

a) Easy for storing, exchanging and managing spatial datasets should be user friendly
b) The spatial databases should provide much functionality for spatial analysis.
c) Database system should be Scalability and efficiency:

Many DBMS have spatial capabilities are selected in benchmark. i.e. MySQL, Oracle Spatial, PostGIS, and IBM-DB2 Spatial Extender. MySQL and PostGIS are open source and free available to use, while Oracle Spatial and IBM-DB2 are commercial databases that require a license to use.

MySQL is a popular open source DBMS having various storage engines i.e. MyISAM with spatial indexing and InnoDB with foreign key references.

PostGIS, extension to the PostgreSQL database having ability to store spatial datasets and provides high speed spatial indexing using GiST or R-Tree indexes.

IBM DB2's Spatial Extender, uses a plan metric model to speed up analysis time and provides a 3-tiered grid spatial index enhancing traditional hierarchical B-tree index provided by DB2 to allow applications to locate and query the M-D geographic data quickly.

A mature commercial spatial database Oracle Spatial facilitates the storage, retrieval, update and query of collections of spatial features in one database and it uses R-tree index to optimize the ability of spatial queries.

A. Benchmark Dimensions:

a. Supports To Ogc Simple Feature Specification Ogc:

Feature specification describes a hierarchy of geometry types, including Point, Curve, and Surface. The above four databases support almost all of the geometry types except Polyhedral Surface. OGC SFS defines 51 routines based on above geometries for spatial analysis, i.e. determining relationships of overlap, touches, buffer, and point-in-polygon.

IBM DB2 supports almost all the routines except three routines on polyhedral surfaces (ST_Geometries, ST_NumSurfaces, ST_SurfaceN) for spatial analysis. PostGIS provides well support for OGC SFS, but not as IBM DB2.

Different with Oracle Spatial, DB2 Spatial extender and PostGIS, all defines a series of its own routines; extent limits the data level interoperability with PostGIS and DB2.

b. Support For Spatial Operation:

Accurately illustration the efficiency databases, two requirements should be satisfied:

First, the efficiency of various databases relies on hardware configurations of computers. i.e. faster CPU frequencies and large memories; The first requirement is to eliminate the influence of hardware to the performance of databases and reliance of databases to hardware is eliminated.

The second requirement is the SQL statements must be consistent on a certain query i.e. calculating the intersection of a polyline dataset and a polygon dataset, using “count(*)” or not will affect the speed for backend calculation.

Queries which spread to represent spatial analysis cross different types of geospatial data are selected and these queries are ST_touches, ST_Within, ST_Overlap, ST_Distance, ST_Intersection, ST_Buffer, ST_Centroid, ST_X, ST_Startpoint and ST_Area. Selected queries would be supported by all the databases, but database MySQL is lacking in the serious analysis queries. Eliminating the set of queries not supported by database MySQL would restrict the usefulness of this evaluation,

As mentioned previously, the IBMDB2 Spatial extender supports, many of the function of spatial data management and operation, the complex queries, i.e. “ST_Intersection”, to simple queries as “ST_X”, and then is PostGIS and Oracle Spatial. In contrast, MySQL supports only part of the simple queries. For the queries that MySQL support, the executing time is fast, especially for ST_Touches query. However, the rows returned as query results are much smaller than PostGIS, IBM DB2, and Oracle. So, MySQL gets a high performance as a trade-off from “losing” accurate data, which is a result of the fact that MySQL does not support precise spatial operations.

Minimum Bounding Rectangles (MBR) is used on MySQL instead of the actual geometry to perform these queries. The command for "Touches" is therefore "MBRTouches" which is not compliant with the OGC-SFS. Touches will return "true" only if the boundaries of the geometries intersect but not the interiors. It is less likely that bigger rectangles which contains the smaller geometries will touch only boundaries, and therefore each time the MBR interiors intersects MySQL returns an inaccurate "false".

The opposite behavior which we observed for functions MBR_Within and MBR_Intersect follows the same rationale, since MySQL reported the larger number of inaccurate "true" results for the larger Bounding Rectangles than the true geometries reported by the other three contender databases.

Another phenomenon is that the commercial software like Spatial Extender, IBM DB2 and Oracle Spatial don’t outperform PostGIS as expected. This is because of their performance is more relying on computer h/w configuration and our experiment environment has put restrictions on it.

V. CONCLUSTION

In this paper the characteristics of object-relational query processing and spatial database benchmark has been discussed. ORDBMS evolved due to the need of the users to model complex real world problems in a way that is simple and receptive for the developer. This also put new demands on the optimizer to generate efficient execution plans for the query.

With the increasing of data quantity, as well as the diversification of applications, spatial data mining is very wide application prospect. In this paper, in the process of spatial data classification, give considerable thought to spatial factors on the classification results. Through the design of predicate logic for the expression of spatial relationships, we turn the spatial relationship into the impact factors to the classification, and make classification results more practical than the only object attributes factor. Examples of text fully proved the feasibility of the method.

The representation of DB is a generalization of traditional integrity constraints whereas the time component is peculiar to temporal data only. Also, traditional integrity constraints are single state constraints whereas temporal integrity constraints are multi state constraints which can be synchronous or asynchronous.

VI. REFERENCES

[1] K Sriram, O Borchert,, O Kim, P Gleichmann, and D Montgomery,“A Comparative Analysis of BGP Anomaly Detection and Robustness Algorithms”, Cybersecurity Applications & Technology Conference For Homeland Security, 2009.

[2] A. U. Tansel. “Integrity Constraints in Temporal Relational Databases Extended Abstract”, the International Conference on Information Technology: Coding and Computing, 2004.

[3] R Sion, M Atallah, and S Prabhakar, “Rights Protection for Relational Data”, Ieee Transactions On Knowledge And Data Engineering, 2004

[4] E Bertino, and R Sandhu, “Database Security—Concepts, Approaches, and Challenges”, Ieee Transactions On Dependable And Secure Computing, 2005.

[5] P Bosc,and O Pivert, "About Possibilistic Queries and Their Evaluation", Ieee Transactions On Fuzzy Systems, 2007.

[6] Z Zhou, B Zhou, W Li, B Griglak, C Caiseda,Q Huang, "Evaluating Query Performance on Object-Relational Spatial Databases", 2009.

[7] Z Xiao-dong, Y Chun-cheng,M Ni-na, "Method of Knowledge Representation on Spatial Classification", Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009.

[8] M E SALEH, "Semantic-Based Query in Relational Database Using Ontology", Journal on Data, Information and Knowledge Engineering, 2011.