

A Query Optimization Strategy for Implementing Multi Dimensional Model in Spatial Database System

Animesh Tripathy

Assistant Professor, School of
Computer Engineering
KIIT University
Bhubaneswar, India
animesh_tripathy@hotmail.com

Lizashree Mishra

Research Associate, School of
Computer Engineering
KIIT University
Bhubaneswar, India
lizashreemishra@gmail.com

Prashanta Kumar Patra

Professor, Department of Computer
Science & Engineering
CET
Bhubaneswar, India
pkpatra@chek.com

Abstract-Spatial Data Warehouses (SDWs) combine spatial databases (SDBs) and data warehouses (DWs) allowing analysis of historical data. This data can be queried using Spatial On-Line Analytical Processing (SOLAP). SDW and SOLAP systems are emerging areas that raise several research issues. In this paper, we refer to a different problem existing in SDWs that motivated us to propose a framework - a conceptual multidimensional model able to express users' requirements for SDW and SOLAP applications. We present a different research direction that is important to consider providing satisfactory solutions for SDW and for SOLAP systems. That important area is spatial query optimization. For the past several years, the research on spatial database systems has actively progressed because the applications using the spatial information such as geographic information systems (GIS), computer aided design (CAD) and multimedia systems have increased. However, most of the research has dealt with only a part of spatial database systems such as data models, spatial indexes, spatial join algorithms, or cost models. There has been a little research on the spatial query optimization which can integrate them. Most of the spatial query optimization techniques published until now has not properly reflected the characteristics of the SDBs. This paper presents query optimization strategies which take the characteristics of SDBs into account. The application of standard query processing and optimization techniques in the context of an integrated SDB environment is discussed.

Keywords-spatial database; spatial data warehouse; spatial online analytical processing; spatial query optimization

I. INTRODUCTION

Data Warehouse (DW) applications are designed using a multidimensional view of data consisting of fact and dimension tables. DW data can be dynamically manipulated using On-Line Analytical Processing (OLAP) systems. The latter requires also the multidimensional view of data. Current DWs typically include a location dimension, e.g., store address, client address, or city name. The advantages of using spatial data in the analysis process are well known since visualizing data in space allows revealing patterns that are difficult to discover otherwise [3]. Therefore, spatial databases (SDBs) can give insights about how to represent and manage spatial data in DWs. SDBs allow to store and manipulate spatial objects. The latter correspond to real-

world entities for which the application needs to keep their spatial characteristics. Spatial objects consist of a descriptive component and a spatial component. The spatial component includes its geometry, which can be of type point, line, surface, or a collection of these types. Spatial objects can relate to each other forming topological relationships. They are based on the representation of a spatial extent by a set of points, composed of three subsets: the boundary, which delimits the extent, the interior, i.e., the points within the boundary, and the exterior, i.e., the points outside the boundary. SDBs are typically used for daily business manipulations [6].

The field of spatial data warehouses (SDWs) emerged as a response to the necessity of analyzing high volumes of spatial data. SDWs combine SDB and DW technologies exploiting the capabilities of both systems. Further, Spatial On-Line Analytical Processing (SOLAP) is increasingly gaining the interest of decision-making users. SOLAP systems provide OLAP capabilities of dynamic data manipulation and aggregation referring them to spatial data [1]. However, SDWs raise several research issues, such as the meaning of SDWs, multiple representations of spatial objects, use of multidimensional model that takes into account specific semantics of SDW applications facilitating SOLAP operations.

SDBs have the following characteristics compared with traditional relational databases or object-oriented databases. First, SDBs store non-spatial data as well as spatial data, and a query in the SDBs is a mixed query which contains both spatial subqueries and non-spatial subqueries. Second, the processing cost of the spatial query is very expensive because spatial data is more complex and larger than non-spatial data [5, 7]. Therefore, the spatial query has been processed mostly in two steps, the filter step and the refinement step. Third, most SDBs have spatial indexes for spatial data types and the effect of the spatial indexes is bigger than that of non-spatial indexes. The two-step processing of the spatial query reduces not only the CPU time but also the I/O time because the two-step processing makes it possible to obtain the object identifiers for the candidate objects by accessing only the spatial index [9].

Several SDB systems have addressed the optimization problem for the spatial and non-spatial mixed query in the literature. However, none of the existing optimizers provides

the filter and refinement steps for spatial operators as individual operators [2]. When an input query consists of a spatial subquery and a non-spatial subquery and there is a spatial index for the spatial subquery, the processing order of “filter step – non-spatial operation – refinement step” can be more efficient than that of “non-spatial operation – spatial operation” or “spatial operation – non-spatial operation.”

This paper presents query optimization strategies which take the characteristics of SDBs into account. In this paper we refer to implementation issues of SOLAP operations.

II. RELATED WORK

In the past decade, the database community has extensively studied numerous issues arising from efficient implementation of a SDB, covering data representation/modeling, query languages, index structures, and very importantly, query optimization. Currently, there is no consensus in the literature about the meaning of SDWs. The term is used in different situations, for example, when there are high volumes of spatial data, when the integration or aggregation processes for spatial data are required, or when the decision-making process uses spatial data. Even though it is important to consider all the above aspects in SDWs, what is still missing is a proposal for SDWs with clearly-distinguished spatial elements required for multidimensional modeling. Secondly, applications that include spatial data are usually complex and need to be modeled taking into account users’ requirements. However, the lack of a conceptual approach for DW and OLAP systems joined with the absence of a commonly-accepted conceptual model for spatial applications, make difficult the modeling task. Existing conceptual models for SDBs are not adequate for multidimensional modeling since they do not include the concepts of dimensions, hierarchies, and measures [8].

To the best of our knowledge, very few proposals refer to conceptual modeling for SDW applications. Finally, a proposal for a spatially-extended conceptual multidimensional model should consider different issues not present in conventional multidimensional models.

In the past, much research has been done on query optimization techniques for databases. Optimization efforts mainly concentrated on queries stemming from administrative applications. Recently, databases are increasingly used not only for administrative, but also for technical applications. This is caused by aspects such as huge amounts of data, complex data structures, and the relative importance of other operations. To reduce the heavy burden that operations in technical applications place on the database management system, filters are used. A filter acts as a preprocessor for an operation. The main idea of filters is to reduce the size of the operands. Thus, a filter is used before an operation to reduce the size of its operands (just like semi-joins are used to reduce join-operands in relational queries). With smaller operands, the cost of an operation will be smaller; however, the cost of using a filter has to be taken into account. In addition to using one filter, several filters may be combined into a filter sequence. Each filter in such a sequence will reduce the operand in size or simplify it [4].

III. THE PROPOSED FRAMEWORK

The proposed framework which has been already implemented with spatial dimensions, measure and all SOLAP operations consists of four layers: Internal Layer, Designed Layer, Operational Layer and Display Layer. The framework has been shown in Fig. 1 and the description of these four layers is given below:

A. Internal Layer

The first layer consisting of heterogeneous data sources (databases, flat files, etc.) forms the Internal Layer. According to the user’s request, data are fetched from various data sources using the ETL (extract, transform and load) process.

B. Designed Layer

The second layer is the Designed Layer. After retrieving the data from various data sources, they are kept in DWs. The spatial data transfer standard provides cartographers with a consistent set of terminology and concepts around which data structures can be developed. Spatial data structures are modeled using the transformed data. A spatial data structure is a framework of spatial data, metadata, users and tools that are interactively connected in order to use spatial data in an efficient and flexible way. This helps to acquire process, use, maintain, and preserve spatial data. The data and metadata should not be managed centrally. They should be managed by the data originator and the operations are connected to the various sources. Different types of data warehouse schemas are also constructed in this specific layer. The actual physical structure of a data warehouse is related to a multidimensional data cube which provides a conceptual multidimensional view of data and allows pre-computation and fast accessing of summarized data.

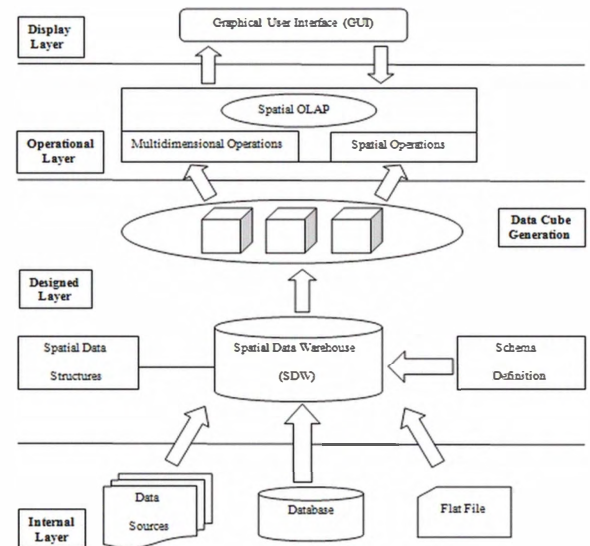


Figure 1. The Proposed Framework.

C. Operational Layer

The third layer is the Operational Layer which performs the spatial OLAP operations on data cubes by integrating both the multidimensional operations and spatial operations. Multidimensional operations and spatial operations are carried out on the data of SDW with the help of multidimensional data views and the precomputation of summarized data which is well suited for analytical processing. Multidimensional operations provide multidimensional data views and spatial operations are functions that form important components of an underlying model that takes input data related to location, performs analysis on it, and assimilates the data to produce output information. These processes are together known as spatial OLAP. So, this layer is responsible for processing all the SOLAP operations on user's request.

D. Display Layer

The fourth layer is the Display Layer. The layer displays the result of user requests through interfaces. It defines the user interface, according to the requirements. When a user gives request, the request is first processed and then displayed in the forms of web, maps, and commands via the user interface with the help of materialized views of the spatial data.

IV. DESIGNED SPECIFICATIONS

A SDW can be constructed using a spatial data cube model/spatial multidimensional database model. A data cube allows data to be modeled and viewed in multiple dimensions. It has two important parameters which is defined by dimension and measure linked together to build a fact table. The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of schemas such as star schema, snowflake schema and the fact constellation. Method diagram for designed requirements composites of mainly five steps: Schema Construction, Schema Definition, Data Cube Generation, Construction of Lattice, and Spatial OLAP operations. Required information is generated only after all these steps are completed properly. All these steps are shown below in Fig. 2.

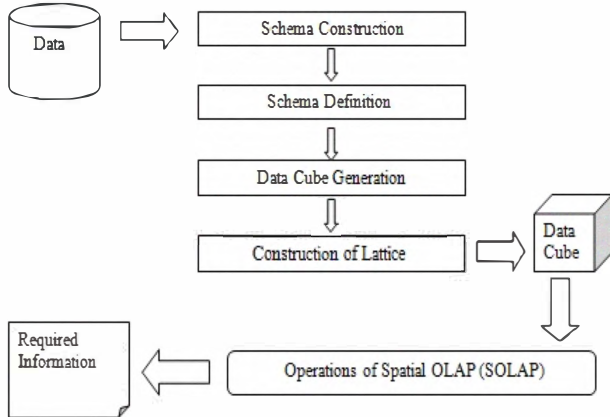


Figure 2. Method Diagram.

V. SPATIAL QUERY OPTIMIZATION

The spatial query has been processed in two steps, the filter step and the refinement step, due to a large volume and high complexity of the spatial data. Here we'll discuss about a query optimization strategy which takes the characteristics of SDBs into account.

We use the select-merge rule of relational algebra optimization rules for combining refinement steps, and the Oid-intersection technique and the Oid-join technique for combining filter steps. We also use the Spatial Object Algebra (SOA) to represent the input query and Intermediate Spatial Object Algebra (ISOA) to optimize the spatial queries.

For example, consider the following mixed query consisting of a spatial operation and a non-spatial operation.

Query 1: Find all hospitals which are inside of the given rectangle (x1, y1, x2, y2) and were completed before 1990.

OQL 1: select h from h in hospitals where h.shape s_inside s_rectangle (x1, y1, x2, y2) and h.comp_date < "90/01/01";

Fig. 3 is an SOA-tree which is generated from the Object Query Language (OQL) parser for the above query. On the spatial attribute "shape" of the class "hospitals", the spatial select operation (S_SELECT) in Fig. 3 is separated into the spatial select filter (SSF) and the spatial select refinement (SSR) operations. Fig. 4 shows the separation. Obviously, SSR is a SELECT operation of the relational algebra because it is generated from a select operation (S_SELECT). Therefore, the query in Fig. 4 can be transformed to the query in Fig. 5 which is in the order of "filter step – non-spatial operation – refinement step". The processing of the original query in the order of Fig. 5 can be more efficient than the order of "spatial operation – nonspatial operation" like Fig. 3 or "non-spatial operation – spatial operation". If the filter and refinement steps are not separated at the logical operator level, only the execution plan like "spatial operation – non-spatial operation" or "non-spatial operation – spatial operation" can be generated.

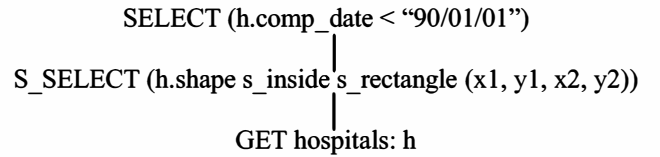


Figure 3. SOA-tree for Query 1.

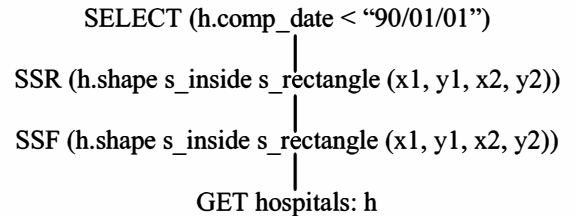


Figure 4. Separation of filter and refinement steps.


```
"select * from MyData where time='M1' and
location='Chennai' order by public_space"; and sql =
"select * from MyData where time='M1' and
public_space='Bank' order by location";
```

VII. ADVANTAGES OF THE PROPOSED FRAMEWORK

The proposed framework is based on an integrated architecture which aims to achieve interoperability through the integration of heterogeneous data sources and it supports all the SOLAP operations. By using SOLAP, users enhance their capacity to explore the underlying dataset once spatial methods incorporated into OLAP ones are used.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we discussed a query optimization technique which took the characteristics of SDBs into account. The main idea is to reflect the two-step processing of a spatial query, which has been applied only in the query execution phase, from the query optimization phase. We showed that filter and refinement operations could be separated at the logical operator level of the query optimization, then the separated filter and refinement operators could be combined with other non-spatial operators or spatial operators in the same level. We have proposed a framework which enables the implementation of a spatial data warehousing and thus validates the concept of a true SDW.

Future works are as follows. First, we plan to implement the query execution module on our proposed framework of SDW and conduct more experiments using real data. Second, we will extend our optimizer to include other SOA operators such as PROJECT, UNION, etc. Third, we will analyze the optimization time and derive algorithms to reduce the optimization time.

Last, since our optimization technique is expected to be more efficient in raster data and image data, we plan to apply the technique to such areas. Spatial data warehousing and spatial query optimization both are still in their infancy and more research on these topics are due.

REFERENCES

- [1] S. Bimonte, A. Tchounikine, and M. Miquel. "Spatial OLAP: Open Issues and a Web Based Prototype." In the Proc. Of 10th AGILE International Conference on Geographic Information Science, Aalborg University, Denmark, 2007.
- [2] Y. Bishr. "Overcoming the semantic and other barriers to GIS interoperability." International Journal of Geographical Information Science, 12(4), 299-314, 1998.
- [3] F. Fonseca, C. Davis and G. Camara. "Bridging ontologies and conceptual schemas in geographic information integration." Geoinformatica, 7(4), 355-378, 2003.
- [4] J. Han, K. Koperski and N. Stefanovic. "GeoMiner: A system prototype for spatial data mining." In Proceedings of the ACM-SIGMOD International Conference on Management of Data (SIGMOD'97), Tucson, AZ (pp. 553-556), ACM Press, 1997.
- [5] D. S. Mackay. "Semantic integration of environmental models for application to global information systems and decision making." SIGMOD Record, 28(1), 13-19, 1999.
- [6] E. Malinowski, and E. Zimányi. "Representing spatiality in a conceptual multidimensional model." In Proceedings of the ACM

Workshop on Geographical Information Systems, Washington, DC (pp. 12-21), ACM Press, 2004.

- [7] H. J. Miller, and J. Han (Eds.), "Geographic data mining and knowledge discovery", (pp. 74-109). London: Taylor & Francis.
- [8] S. Rivest, Y. Bédard, and P. Marchand. "Towards better support for spatial decision-making: Defining the characteristics of spatial on-line analytical processing." Geomatica, 55(4), 539-555, 2001.
- [9] W. Aref and H.Samet. "Optimization Strategies for Spatial Query Processing," Proc. of VLDB, 81-90, 1991.