ZESZYTY NAUKOWE POLITECHNIKI ŁÓDZKIEJ
Nr 1050 ELEKTRYKA, z. 119 2009

TOMASZ MAREK KOWALSKI
Technical University of Łódź
Department of Computer Engineering

TRANSPARENT INDEXING IN DISTRIBUTED OBJECT-ORIENTED DATABASES

Reviewer: **Professor Kazimierz Subieta, Ph.D., D.Sc.**

Manuscript received 18.06.2009

This paper deals with the development of robust transparent indexing architecture for distributed object-oriented database. . The solution comprises management facilities, an automatic index updating mechanism and an index optimiser. The main goal of the work is designing the indexing architecture facilitating processing of a possibly wide family of predicates. Additionally, a technique based on indexing addressing a particularly difficult query optimisation domain, i.e. processing queries addressing heterogeneous resources, has been proposed. The developed algorithms and solutions have been verified in the ODRA OODBMS prototype.

1. INTRODUCTION

Databases are a fundamental feature of many large computer applications. In many cases databases are to be geographically distributed. The size and complexity of such systems require the developers to take advantage of modern software engineering methods which as a rule, are based on the object-oriented approach (cf. UML notation). In contrast, the industry still widely uses relational databases. While the efficiency of them in majority of applications cannot be questioned, many professionals point out their drawbacks.

Query optimisation in object-oriented database management systems has been deeply investigated over last two decades. Unfortunately, this research remains mostly not implemented in currently used OODBMSs due to many reasons: limited query languages, non-implementable methods that were proposed, lack of interest of commercial companies, etc.

A well-known and the most important method of performance improvement known as indexing is investigated in this paper. The research addresses this subject in the context of the Stack-Based Architecture (SBA) [1], which is a theoretical and methodological framework for developing object-oriented query and programming languages. The developed solutions are implemented and tested in the ODRA OODBMS prototype [2][3][4] that is based on SBA and its own query language SBQL (Stack-Based Query Language).

2. SHORT STATE OF THE ART OF INDEXING IN DATABASES

The general idea of indices in object-oriented databases does not differ from indexing in relational databases [5][6][7]. The most characteristic property of the database indexing is transparency. A programmer of database applications does not need to be aware of the indices existence as they are utilised by the database engine automatically. This is usually accomplished by a query optimiser that automatically inserts references to indices into a query execution plan when necessary. The second important aspect of transparency concerns maintaining cohesion between existing indices and the data that is indexed. Data modifications are automatically detected and corresponding changes are reflected in indices. This process is called automatic index updating.

In the object-oriented database domain the research into indexing has been mainly focused on a path expression processing and an inheritance hierarchy inside indexed collections [8][9]. Some papers propose generic approaches to provide automatic index maintenance transparency [10]. However, there is no information that these proposals have been actually incorporated in commercial or open source database products.

Indexing is also an important subject in a distributed environment. The most of research concerns development of various distributed index structures and global indexing strategies. In databases, the most advanced solutions are based on static index partitioning. They are implemented in leading object-relational products. Nevertheless, an index key definition is limited to expressions accessing data from only one table. In the research literature no formalised global optimisation methods based on indexing for processing queries involving heterogeneous resources, were found.

The analysis of the state of art unambiguously indicates that the development of indexing methods and architectures that are dedicated to distributed object-oriented databases, is still a relevant and challenging subject.

3. ORGANISATION OF INDEXING IN OBJECT-ORIENTED DATABASE

The proposed organisation provides transparent indexing using single or multiple-key indices. The first important element of the developed architecture is precisely defined *indices management facilities* and convenient syntax for an index call to be used in query optimisation.

The second part composing indexing in OODBMS is the author's generic approach to automatic index maintenance shown in Fig. 1.

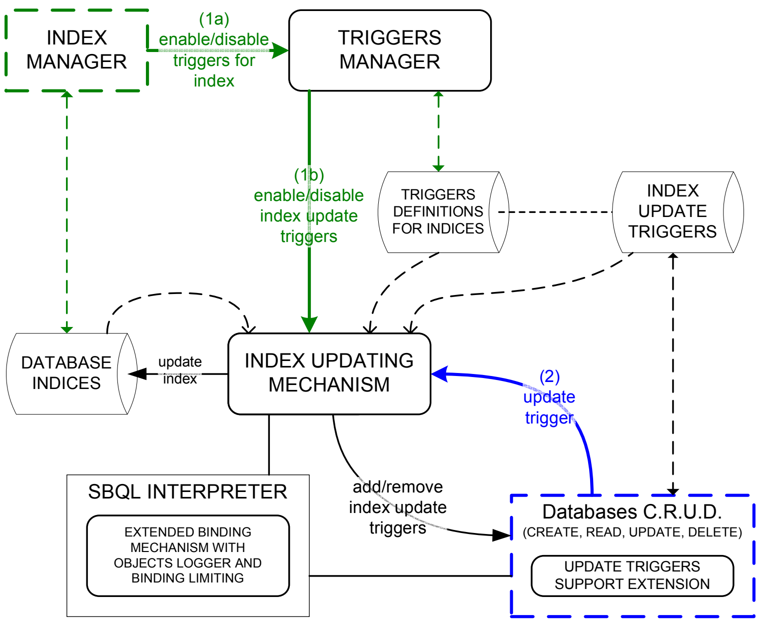

Fig. 1. Automatic index updating architecture

It is based on *index update definitions* assigned to indices and associated with them *index update triggers* assigned to the objects participating in indexing. It applies to selection predicates based on arbitrary, deterministic and side effects free expressions consisting of e.g. path expressions, aggregate functions and class methods invocations (addressing inheritance and polymorphism).

The third element of the developed architecture is a set of algorithms, optimisation methods and rules composing the *index optimiser*, i.e. the module responsible for detecting parts of a query that can be substituted with an index call and performing appropriate query transformations. Its schema is depicted in Fig. 2.

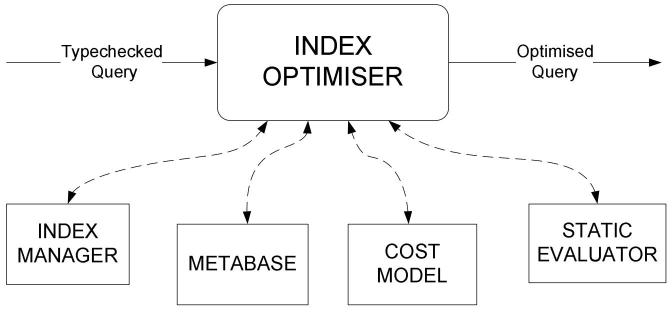

Fig. 2. Schema of the *index optimiser*

The example optimisation scenario is shown using a query retrieving persons with a surname "KOWALSKI" who are 28 years old.

```
Person where ((surname = "KOWALSKI") and (age = 28))
```

The *index optimiser* applies the *idxPerAge* index which retrieves *Person* objects according to *age* attribute and rewrites a query to the following form:

```
$index_idxPerAge(28) where surname = "KOWALSKI"
```

First, the predicate *age* = 28 is selected and removed. The *index optimiser* replaces the ***where*** left operand (*Person*) with an index invocation exactly matching the removed predicate. This transformation preserves semantic equivalence and reduces the number of objects evaluated by the ***where*** operator.

4. VOLATILE INDEXING TECHNIQUE

A particularly difficult query optimisation domain concerns processing queries addressing heterogeneous resources. A *volatile indexing technique* proposed by the author is a significant step in this matter. This solution relies on the developed indexing architecture. Additionally, it can be applied to data virtually accessible using SBQL views.

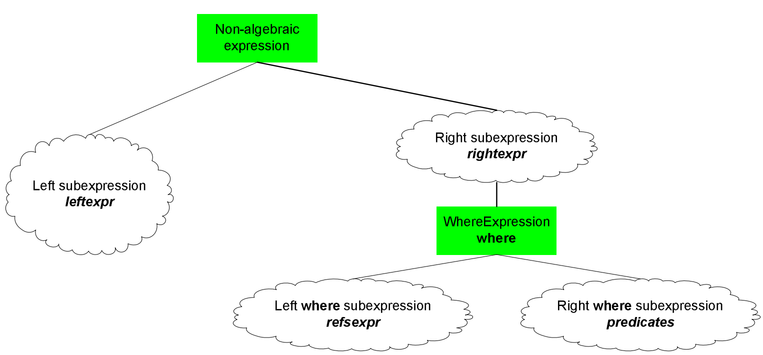

Fig. 3. Query suitable for applying a *volatile index*

In contrast to regular indices, a *volatile index* is materialised only during a query evaluation. Therefore, efficacy of this technique shows when the index is invoked multiple times, which mainly concerns processing of complex and laborious queries. Such situation can occur when the optimised ***where*** clause is situated on the right side of a non-algebraic operator. It is presented in Fig. 3. When the *leftexpr* expression returns a collection the *rightexpr* containing a *where* clause is multiply evaluated against each collection result. It is assumed that there exists an index defined on objects returned by the *refsexpr* expression. *Predicates* expression should contain selection predicates defining key values for a given index, which are dependent on a given non-algebraic operator; so, the index key would be context dependent.

Practical verification of the proposed technique has been conducted on an example query addressing heterogeneous schema with the transparent integration of RDBMS resource into ODRA distributed database repository. The *volatile indexing technique* approach to query optimisation resulted in more than 40 times better processing performance.

5. CONCLUSIONS

The designed indexing architecture provides the assumed level of transparency in the distributed and homogeneous environment with horizontal fragmentation. The architecture comprises an optimisation module that is able to employ indices in a query, automatic update of indices in response to a modification of corresponding data and an administration module for organising and managing indices.

Additionally, a *volatile indexing technique* has been proposed and proved useful. It enables taking advantage of the developed indexing architecture and omits troublesome issue of the automatic index maintenance in processing a specific family of queries addressing heterogeneous resources.

ACKNOWLEDGEMENTS

This research work is funded from the science finances in years 2008/2009 as a research project nr N516 383334.

REFERENCES

[1] **Subieta K.**: Stack-Based Approach (SBA) and Stack-Based Query Language (SBQL). http://www.sbql.pl , 2008.

[2] **Adamus R., Kowalski T.M., Subieta K.** et al.: Overview of the Project ODRA. Proceedings of the First International Conference on Object Databases, ICOODB 2008, Berlin, ISBN 078-7399-412-9, pp. 179-197.

[3] **Kowalski T.M.** et al.: Local and Global Indexing Strategies and Data-Structures in Distributed Object-Oriented Databases. SiS 2006, Łódź, Poland, 2006, pp. 153-156.
[4] **Kowalski T.M.** et al.: Optimization by Indices in ODRA. Proceedings of the First International Conference on Object Databases, ICOODB 2008, Berlin, ISBN 078-7399-412-9, pp. 97-117.
[5] **Chaudhuri S.**: An Overview of Query Optimization in Relational Systems. Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, Seattle, Washington, United States , pp. 34-43, 1998.
[6] **Elmasri R., Navathe S. B.**: Fundamentals of Database Systems. 4th Edition, Pearson Education, Inc, publishing as Addison-Wesley, 2004, ISBN 0-321-12226-7
[7] **Jarke M., Koch J.**: Query Optimization in Database Systems. ACM Computing Surveys 16(2), 1984, pp. 111-152.
[8] **Bertino E.** et al.: Definition and Analysis of Index Organizations for Object-Oriented Database Systems. Information Systems, v.23 n.2, p.65-108, April 1, 1998
[9] **Maier D., Stein J.**: Indexing in an object-oriented DBMS. In Proceedings on the 1986 international Workshop on Object-Oriented Database Systems, IEEE Computer Society Press, pp. 171-182.
[10] **Henrich A.**: The Update of Index Structures in Object-Oriented DBMS. Proceedings of the CIKM'97, Las Vegas, Nevada, November 10-14, 1997. ACM 1997, ISBN 0-89791-970-X: pp. 136-143.

PRZEZROCZYSTE INDEKSOWANIE W ROZPROSZONYCH OBIEKTOWYCH BAZACH DANYCH

Streszczenie

W artykule opisano opracowaną zaawansowaną architekturę przezroczystego indeksowania dla rozproszonej, obiektowej bazy danych. Na rozwiązanie składa się moduł zarządzania, mechanizmy automatycznej aktualizacji indeksów oraz optymalizator stosujący indeksy. Głównym celem pracy jest opracowanie architektury indeksowania, która wspomagałaby przetwarzanie możliwie szerokiej rodziny predykatów. Dodatkowo, zaproponowano technikę opartą o indeksowanie, która dotyczy szczególnie trudnej dziedziny optymalizacji zapytań, tj. przetwarzania zapytań odnoszących się do heterogenicznych zasobów. Opracowane algorytmy i rozwiązania zostały zweryfikowane na prototypowej w obiektowej bazie danych ODRA

Promotor: prof. dr hab. inż. Kazimierz Subieta
Recenzenci pracy doktorskiej:
1. dr hab. inż. Adam Pelikant, prof. PŁ
2. dr hab. Krzysztof Stencel, prof. UW