Mathematical & Computational Applications, Vol. 8, No. 3, pp. 263-269 2003

MULTILEVEL ARCHITECTURE FOR MULTI DIMENSIONAL DATA BASE

M.Ali Salahli
Chanakkale OnSekiz Mart University, Chanakkale, Department of Computer Engineering
(msalahli@comu.edu.tr)

Abstract- A Multidimensional Data Base is an essential element of decision support, which allows to process complex queries. In this paper, a multilevel distributed Data Model for MDDB is presented. Metadata for MDDB on relations is introduced. To improve efficiency of query processing, fuzzy **cache fact table is proposed.**
Keywords- Multidimensional Data Base, Data Model, Multilevel Star-scheme, Fuzzy cache fact table.

1. INTRODUCTION

The fundamental problem of a MDDB design is to determine the Data Model Structure for the Data Base, which improves the performance of the MDDB to answer end users queries. The most relational Data Models for Multidimensional Data Bases are built on the star scheme. A star scheme includes a fact table and dimension tables, where every dimension table contains information that belongs to the dimension itself. Fact table connects all dimensions and contains information on the attributes of interest for the intersection of all the dimensions. As a consequence, fact table consists of large number of the records and queries processing time is increased. To reduce this time, several techniques and algorithms have been proposed.

In this paper, we address a distributed multilevel Data Model for MDDB. The main contribution of this paper to propose a distribution multilevel star-schemes concept and a relational structure of the Metadata. To process frequently used queries, the principle of fuzzy caching of fact tables is presented.

The paper 's constructed as follow. The brief analysis of the related works are discussed in section 2. The Distributed Data Model for Multidimensional Data Base is presented in the next section. In the section 4 the Metadata structure of the MDDB is given. Also, the paper includes the conclusion.

2. RELATED WORKS

A different approach is used to develop a Multidimensional Data Base and Data Model for MDDB are discussed in the related literature. [1-6]. In [1] some data warehouse architectures are reviewed. Evolutionary approach to Integrated Data Warehouses architecture is presented. Data Warehouse Data Model is given. In [2] the logical approach to multidimensional Database is presented. Design problems of multidimensional Data Model are discussed in [3].

An Object Oriented Multidimensional Data Model for MDDB is given in [4-6]. The formal definition of conceptual Data Model and its components are presented. Formal description of the MDDB model has been discussed in [4]. The framework for the definition of attribute hierarchies in star-schemes is presented. The Multi Dimensional star-scheme in relational MDDB is discussed. Metadata for Data Warehouse system on object-relational Data Base has been presented in [5]. The classification of Meta Data for Star-scheme in object relational Data Warehouse is given. Processing relational queries in MDDB is discussed in [7].

There are different approaches to gain an acceptable level of performance in Data Warehouse. In particularly, problems of view selection and maintenance, query equivalence for using the views are discussed in [8]. The technique, reducing the number of view and solution space by selection and materializing view in MDDB are given. However, the problems of application of cache technique for Data Warehouse have been discussed in a few works [9,10]. In particularly, the global cache manager is described in [10]. The algorithm, to generate efficient cache allocation scheme for multiple queries is proposed.

3. DISTRIBUTED DATA MODEL FOR MULTIDIMENSIONAL DATA BASE

Database is a data repository that is necessary to solve various management problems, such as decision support and decision-making, data analyzing and reporting. A MDDB consists of huge amounts of information to answer the complex management queries. In most MDDB this information is organized following the star-scheme, which includes fact table and dimension tables. Each dimension table contains information, is belonging to the dimension itself. The fact table contains information on the attributes of interest for the intersection of the dimension tables [fig. 1].

In this section, we propose the distribute Data Model for Multidimensional data Base. The idea of distributed Data Model is based on the concept of distribution information systems. If an application domain has distributed properties, such architecture is more reasonable than traditional centralized architecture for MDDB.

Our Data Model has a hierarchical structure. To simplify the description, we assume that the numbers of the hierarchy level are 2. For many applications it is sufficient.[fig.2].

The MDDB Data Model is constructed based on sets of distribution star schemes.

Definition 1. A distribution star-scheme is a set of star-schemes $\{S_1^1, S_2^1, \ldots, S_n^1, S^2\}$, where S_i^1 $(i \in N)$ is a star –scheme in the lower level of the hierarchy, and is called as sub star-scheme; S^2 is a star scheme of the second level, and is called as superstar-scheme.

Definition 2. Superstar scheme is a set of tables $\{D_1^2, D_2^2, \ldots, D_k^2, F_1^1, F_2^1, \ldots, F_n^1, F^2\}$

where D_k^2 $(k \in K)$ is a dimension table, belonging to the scheme S^2;

F_n^1 $((i \in N)$ -is a fact table, belonging to the scheme S_n^1; F^2 is the fact table from the S^2.

According to the functional hierarchy of the application area, the fact tables form the class structure. Consequently, the domain of the tuple identifiers of the table F^2 forms a class of the object identifiers $\{d_1^1\text{-id}, d_2^1\text{-id}, \dots, d_n^1\text{-id}\}$, where d_n^1-id determines a tuple of the fact table F_n^1 and is a identifier of the lower level objects, associated with the dimension tables. For example, a distribution information system for University Management can consist of following hierarchy: *university- faculty-department.*

The fact table F^2 in the *university* level consists of *faculties'* identifiers (*university* is a class of object *faculty*) . Each *faculty* identifier is a class of the department identifiers. The *department* identifiers determine a tuple of the *faculty* dimension table.

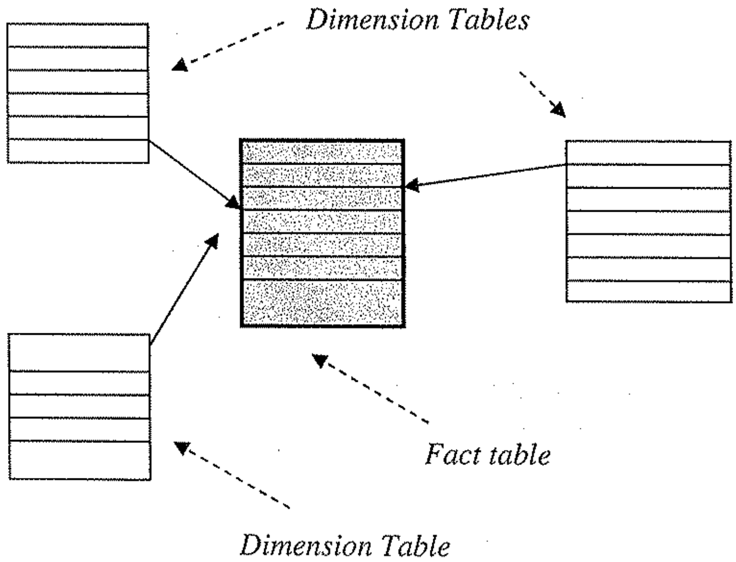

Figure1. Star Schema for Multidimensional Database

According to the functional hierarchy of the application area, the fact tables form the class structure. Consequently, the domain of the tuple identifiers of the table F^2 forms a class of the object identifiers $\{d_1^1\text{-id}, d_2^1\text{-id}, \dots, d_n^1\text{-id}\}$, where d_n^1-id determines a tuple of the fact table F_n^1 and is a identifier of the lower level objects, associated with the dimension tables. For example, a distribution information system for University Management can consist of following hierarchy: *university- faculty-department.*

The fact table F^2 in the *university* level consists of *faculties'* identifiers (*university* is a class of object *faculty*) . Each *faculty* identifier is a class of the department identifiers. The *department* identifiers determine a tuple of the *faculty* dimension table.

Every measure value may be derived from aggregation of the appropriate value of measure from the fact tables of the lower level hierarchy. It allows speeding up rolling-up and drilling down operations.

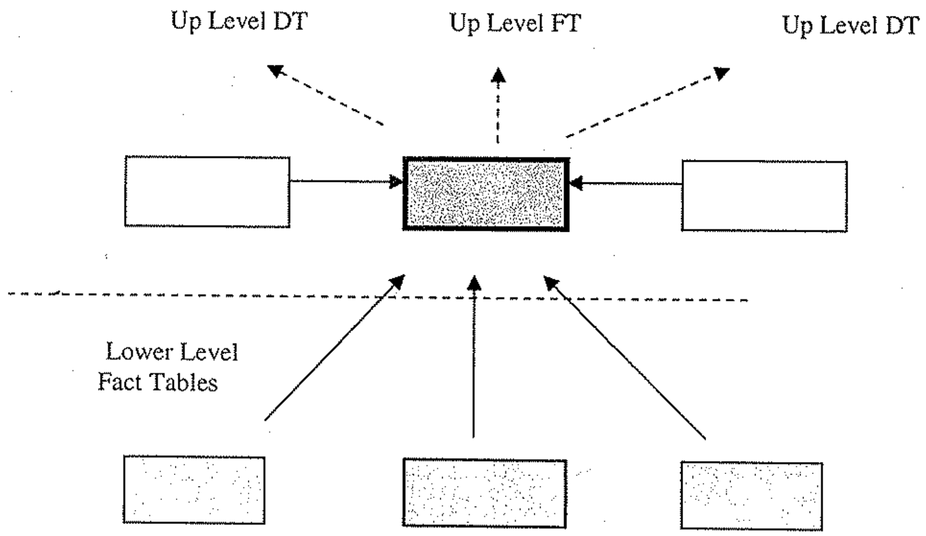

Figure 2. Hierarchical Star Scheme Structure

4. THE STRUCTURE OF THE METADATA

In [5] it is given generally classifications Data Warehouse Metadata as follows:

– Metadata for description of the source data and any changes, that were made to the data.

– Metadata for define the data store in the data warehouse.

– Metadata for query management.

To describe the metadata and relations between the metadata our model uses the following relations:

Metadata description (md_id , name, type, h_level);

Attribute description (md_id, attr_id);

The relation contains information about of all data in the Data Model. Each data is identified by unique identifier md-id. Data are characterized by name, type (dimension table, fact table, cache table, attribute, measure, query). h_level indicates the level number, where the data is used.

To represent the attribute hierarchy we use the tree structure. A root of the tree corresponds to an aggregated attribute from the highest level of the hierarchy. A leaf of the tree corresponds to an elementary attribute item. The nodes from same level are called brothers. Only one of the brothers (first brother) is connected with the farther node. Relations between a farther and the remaining sons are realized via the first brother. To express the tree structure of an attribute hierarchy we use 2 relations :

Tree_description(md_id, farther-id, son_id);

Brother_description(m_id, f_brother_id)

To describe the metadata for the measures, we introduce the relation *Measure-description(m_id, md_id,).*

Relationships between metadata model and Data Base Model are realized via relation *Tuple_description(f_id, tuple_id, m_id).*

The relation contains information about of the location of the measure in the fact table.

A cache fact table has been added to the Data Model to improving the performance characteristics. Different caching algorithms are widely used in operating system and database system fields. A simple replacement algorithm is LRU [11]. According the algorithm the cache manager determines least recently used tuple in the cache table and replaces it with a last used tuple from the fact table.

We present the approach, based on application of fuzzy set theory. The main idea of the approach is following.

The cache table is parted to several sections. Every section obtains information about of the queries, which usage frequency may be defined with some linguistic term, such as rarely, seldom, sometime and e.g. [fig.3]

The set of these terms forms the fuzzy variable *f_frequency*. The elements of the set are named as fuzzy value of the *f_frequency.*

f_frequency={rarely, seldom, sometime, often, usually}

There exists some presentation of the fuzzy variables [12]. We use trapezoidal presentation, where every fuzzy value is presented with four crisp numbers. These numbers, as attribute of the value are included in the table *Fuzzy_frequency_value.*

Initially, the table consists of expected queries identifiers and corresponding expected frequency of the queries. Then the cache table is formed using the data from the tables *Fuzzy_frequency_value* and *Query frequency.*

The algorithm for looking up information about of the users queries from cache table as follows.

Step 1. Determining the query identifier.

Step 2. Determining the expected fuzzy frequency and value its membership function, using the table *Fuzzy_frequency_value* and *Query frequency.* If the query identifier does not exist in the table, then adding the necessary information to the table.

Step 3. Searching the query information in the part of the cache table, where corresponding value of the membership function of the query is maximal. If the necessary information is not found in this part, searching is continued in the part, corresponding to the next from maximal value of the membership function.

4. CONCLUSION

In this paper, the structure of Information Management System based on Distributed Data Warehouses has been introduced. To process frequently used queries a principles of caching of fact tables is supposed. The fuzzy approach to building of cache fact table, allowing to speed up of query processing is proposed. The structure of a Metadata level for

Data Warehouse has been introduced. The main contributions of this paper are used in design of the Distributed Information System of Chanakkale On Sekiz Mart University.

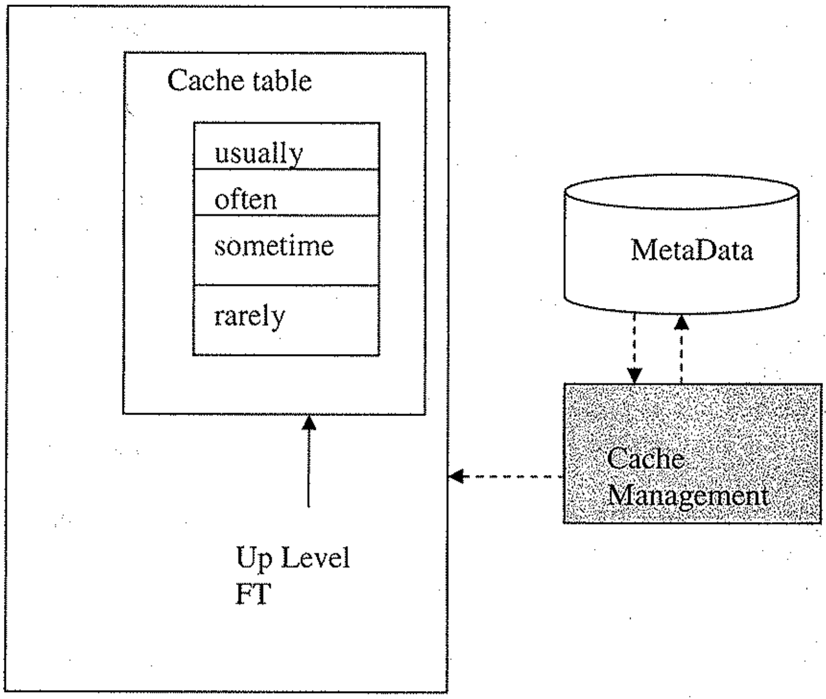

Figure 3. Fuzzy Caching of Query Searching

REFERENCES

1. [1] Jose Samos, Felix Saltor, Jaume Sistac, Agusti Bardes, Data Base Architecture for Data Warehousing: An Evolutionary Approach, *Data Base and Expert Systems Application, DEXA'98*, Springer,1998, vol. 1460.
2. [2] Luca Cabibbo, Riccardo Torlone. A Logical Approach to Multi Dimensional Data Bases, *EDBT' 98*, Springer,1998, vol.1377.
3. [3] M.Gussens, L.V.S.Lakshmanan, A Foundation for Multidimensional Database, *Proc. Int. 33rd Int. Conf. On Very Large Data Bass*, 1997
4. [4] Thanh Binh Nguyen, A Min Tjoa, and Roland Wagner, *An Object Oriented Multi Dimensional Data Model for OLAP*, White Paper.
5. [5] Thanh N. Huynh, Oscar Mangisengi, A Min Tjoa, Meta Data for Object - Relational Warehouse, *Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)*, Stockholm, Sweden, 2000.
6. [6] Joseph M. Firestone, *Object-Oriented Data Warehousing,* White Paper, No Five, March 27,1997

7. [7] Volker Markl, Processing Relational OLAP Queries with UB-trees and Multi Dimensional Hierarchical Clustering, *Proc. Of the Int. Workshop on Design and Management of Data Warehouses (DWDW'2000),* Stockholm, Sweden, 2000.
8. [8] Elena Barabis, Stefano Paraboschi, Ernest Teniente. Materialized View Selection in a Multi Dimensional Data Bases, *Proc. of the 23rd VLDB Conf.*, 1997.
9. [9] P.Deshpande, K.Ramasamy, A.Shunka, J.F.Naughton. Caching Multi Dimensional queries using chunks, *Proc. ACM SIGMOD, Int. Conf. On Management of Data, 1998.*
10. [10] S.Jin,X.Sun, Global Cache Management for Multi-class workloads in Data Warehouses, *CİSE'98,* Springer,1998, vol.1413.
11. [11] Silberschadz Galvin, Operating System Concepts, Addison Wesley Publishing Company, 1998.
12. [12] R.Aliyev, R.Aliyev, *Soft Computing, p.1,* Baku, ASOA, 1998 (in Russian).