A cube model approach for Data warehouse

Zuyi Chen [1, a], Taixiang Zhao [1,b]

[1]Department of foundation First Aeronautic Institute of the Air Force Xinyang, China

[a]zychen7410@163.com, [b]ztx304@126.com

Keywords: Data warehouse; Data cube; OLAP; Data models

Abstract: The grain preservation feature guarantees that the translated multidimensional model will maintain cohesive granularity among the entities. Many data warehouse systems have been developed recently, yet data warehouse practice is not sufficiently sophisticated for practical usage. Most data warehouse systems have some limitations in terms of flexibility, efficiency, and scalability.In particular, the sizes of these data warehouses are forever growing and becoming overloaded with data, a scenario that leads to difficulties in data maintenance and data analysis. This research focuses on data-information integration between data cubes. To deal with the handling of increasing data volume in data warehouses, we discovered important inter-relationships that hold among data cubes, that facilitate information integration, and that prevent the loss of data semantics.

1. Introduction

The function of a data warehouse is to effectively integrate operational databases into an environment that facilitates strategic use of data that, in turn, improves the productivity of a decision-maker through consolidation,conversion, transformation, and integration of operational data. In designing a data warehouse, one would first integrate into one another various data sources from an enterprise's heterogeneous databases.

Whenever data from multiple sources has to be consolidated, developers must analyze the structure and the content of the source before defining the rules for merging. To execute these rules, researchers in the field must develop a satisfactory process. The integration should ensure results that exhibit data consistency across the entire enterprise. Ideally, end user should be able to access data from the data warehouse without knowing either where data resides or the form in which it is stored.

The architecture of data warehouses falls into three categories [1,4,5]:

1.Virtual view approach: The repository of the data warehouse contains only the data schema, and the local databases store the physical data. This approach uses query pre-processing and query shipping to answer queries that queries make against the integrated view. The disadvantage of this architecture is its poor performance.

2. Materialized view approach: The repository of the data warehouse contains the data schema and the physical data. It collects all relevant information in the data warehouse. The disadvantage of this architecture is the difficulty it poses in the managing and the maintaining of a huge data bank.

3.Datamart approach: This approach extracts data from a primary data warehouse for Datamart application.

The extraction has a special purpose. Hence, the repository of the data warehouse contains the subject of information, which is in Datamart format. The disadvantage of the architecture is that a data warehouse contains only limited knowledge. This approach cannot manage huge amounts of data.

Most current data warehouse systems, such as theMicrosoft OLAP server, the Oracle OLAP server, Sybase IQ, and Business Objects, use the third architecture as their data warehouse approach. And most of these commercial software tools use data cubes to represent Datamart, and the use has a special purpose. The data type of the data cube is a multidimensional matrix that lets users explore and analyze a collection of data from many different perspectives and usually from several dimensions at once.

Use of the Datamart approach for the implementation of an enterprise's data warehouse will develop many cubes, each cube is an independent data aggregation. Because a less semantic relationship has been defined between each of the data cubes, they become isolated bits of information[6]. Users retrieve the knowledge from one single angle and not from a global view; therefore, problems like data duplication, inconsistency,and query integrity could occur.

2. Content management

Managing the content of a data warehouse is a daunting task. Locating and acquiring the data needed to produce the types of information described above are significant challenges. Integrating the acquired information may be even more challenging. Modern organizations use a wide variety of distributed information systems to conduct their day-to-day business. These operational systems draw data from a variety of databases that operate on different hardware platforms, use different operating systems and DBMSs, and have different database structures with varying structural, conceptual, and instance level semantics.Existing practice successfully addresses many of the hardware, operating system,DBMS,and structural heterogeneities associated with such systems. However, major challenges remain for data warehouse content management. These include identifying and accessing the appropriate data sources, coordinating data capture from them in an appropriate timeframe, assuring adequate data quality, and integrating instance level data.A data warehouse serves as a repository for data extracted from diverse operational information systems and acquired from external sources. The extract, transform, and load (ETL) functions in a data warehouse are considered the most time-consuming and expensive portion of the development lifecycle[2]. These processes are concerned with the extraction of data from legacy systems and external sources, the transformation and pre-processing necessary to produce useful, integrated data, and the transportation of the data into the actual data warehouse structures.Often operational systems are not designed to be integrated and data extracts must be performed manually or on a schedule determined by the operational systems. Furthermore acquired data from external sources is rarely in a form conducive to integration.

3. Generating association rules

3.1 A multi-layered cache mechanism

The raw data is stored in dimension, fact, and aggregate tables. Information related to each dimension is stored in a separate table. The fact table contains information about measures such as profit and sales, along with keys relating it to the dimension tables. The fact, dimension and aggregate tables can be queried to obtain the measures at different levels of dimensional hierarchies.

Initially, a mobile user is presented with several association rules displayed in the mobile client device, and after selecting a rule of interest, the user may request the data represented by this association rule.

This request is forwarded to the middle tier server as an OLAP query expression and the query result is returned back to the mobile client. Our multi-layered cache mechanism stores the association rules and their correlated query details. These caching layers correspond to three different types of information that are linked together: (i) the generated association rules, (ii) the corresponding multidimensional query expressions and (iii) the actual query results.

3.2 Part assembly

In our system, the user can view the data at various levels of dimensional hierarchies. As an example, let us assume that a decision maker has drilled down the product dimension to the dDrinksT category. Next, let the user now look at the part of the cube where region is dSouth

WestT. The query corresponding to the first request is modified to group the dimensions for year and product subcategory and slice the previous cube for sales region dSouth WestT. Taking this approach we identify all possible multidimensional queries, that will be used to gather the data for data mining purposes.

At times it is important to assemble the components or parts of an organization or item. The assembly of the components leads to an understanding of the relationships between components.

A bridge dimension is used to capture and traverse the relationships between leaves (or nodes) in a hierarchical tree. The hierarchy relationship is implemented via a bridge table that defines the relationship of each node in the hierarchy to the other nodes in the hierarchy. Kimball recommends this approach to represent organizational hierarchies and manufacturing parts explosion hierarchies [3].

A bridge table is also used to solve the problem of multi-valued dimensions. For example, in the financial domain an account may be associated with two or more people thus requiring the linking of those people to a single account. Creating a bridge table between the account and customer dimension can mitigate the multivalued attribute.

4. Access control and audit (ACA) model

Access control is not a complete solution for securing a system [7] as it must be coupled with auditing. Auditing requires the recording of all user requests and activities for their later analysis. Therefore, in our approach, we take both concepts into consideration for their integration in the conceptual MD modeling design. Access control models are typically composed of a set of authorization rules that regulate accesses to objects. Each authorization rule usually specifies the subject to which the rule applies, the object to which the authorization refers, the action to which the rule refers, and the sign describing whether the rule permits or denies the access.In order to regulate access to objects in a MD model, we have considered the Mandatory Access Control model (in the form of multilevel security policies),which is based on the classification of subjects and objects in the system. Therefore, our access control and audit model allows us to specify sensitivity information assignment rules for all elements of MD models(facts, dimensions, etc.), which define static and dynamicobject classification. Moreover, our model allows us to define authorization rules that represent exceptions to the general multilevel rules, where the designer can specify different situations in which the multilevel rules are not sufficient. Finally, a set of audit rules, which represent the corresponding audit requirements, can be included in the model.

5. Conclusions and future work

In this paper, we proposed a methodology, which combines OLAP technology with association rules for decision makers using mobile devices. We addressed the problem of navigating through a very large search space with very a low capacity input/output device. Our approach prunes the search space by providing the decision maker insights into the patterns in the data through association rules. We described a multi-tier architecture and a multi-layered caching mechanism. The mobile cache reduces wireless data exchange between the mobile client and the middle-tier server. The server cache further decreases the amount of data processing time by keeping a copy of the requested data in the server cache, and minimizes communication with the data warehouse. We demonstrated the benefits of our methodology for rule-based cube exploration and caching mechanism through experimental results.

We continually try to improve on our work and minimize its limitations. First it only considers association rules, which, as we have explained in the paper, is at the most appropriate level for most decision-making situations. Although the framework can accommodate data mining methods such as clustering and classification, the paper does not address them. Other useful techniques for decision makers include outlier detection and visualization, which are also beyond the scope of this paper.

References

[1] M.Boehnlein, A.U.Ende, Business process oriented development of data warehouse structures, in: Data Warehousing (DW 2000),Friedrichshafen, Germany, November 2000, pp. 3 – 21.

[2] F.Boufares, S.Hamdoun, Integration techniques to build a data warehouse using heterogeneous data sources. Journal of ComputerScience (Special Issue), 48 (2005) 48 – 55.

[3] R.Barzilay, Information fusion for multi-document summarization: paraphrasing and generation, Ph.D. Thesis, Columbia University,2003.

[4] H.Wache, T.VSgele, U.Visser, H. Stuckenschmidt, G. Schuster, H.Neumann, S. Hfibner, Ontology-based integration of information a survey of existing approaches. The IJCAI Workshop on Ontologies and Information Sharing, 2001.

[5] D.L.Yang, M.L.Huang, M.C. Hung, Efficient utilization of materialized views in a data warehouse, in: Proceedings of the Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, 2002, pp. 393 – 404.

[6] J.Yang, K. Karlapalem, Q.Li, A framework for designing materialized views in data warehousing environment, in: Proceedings of the 17th International Conference on Distributed Computing Systems, Baltimore, MD, USA, 1997, pp. 458 – 465.

[7] J.X.Yu, X. Yao, C.H.Choi, G.Gou, Materialized view selection as constrained evolutionary optimization, IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews 33 (4) (2003) 458 – 467.