# Study on Building and Applying Technology of Multi-level Data Warehouse

Lihua Song, Ying Zhan, Yebai Li

College of Computer Science,North China University of Technology,Beijing, China

e-mail: blackzhanying@qq.com

**Keywords:** Multi-level Data Warehouses; Extract; Incremental Update； E- Business

**Abstract.** Based on multi-database management systems application, we use multi-level data warehouses. This paper proposes an incremental method to update the data set of result and summarizes the two transmission strategies. Verify that the data-driven policy through experimental simulation platform environment analysis and draw the feasibility and effectiveness of this method is more suitable platform applications conclusion.

## Introduction

In the background of fast-growing information, data shows the characteristics of massive, distributed and heterogeneous. It makes the centralized data warehouse processing capacity in data analysis is increasingly limited. Because of distributed data warehouses have the features of low cost of maintenance, data integrity, high tolerance against system failures[1], it's more competitive for some special cases which include bank and e-commerce platform.

In recent years, Hive is the mainly technology to be used to build distributed data warehouses[2], but Hadoop is not yet available analysis tools. In order to avoid complex development work in the application presentation layer, we use the database technology and combine with the open source analysis and presentation tools(e.g. Mondrian and JPivot). Related research on multilevel data warehouses structure for typical distributed data warehouses form(including global warehouse and local data warehouses) is as follows: Paper [3] presented double channels view updating algorithm to maintain multiple views on line and parallel realize OLAP query for ensuring data consistency and improving query efficiency. Paper [4] proposed a model of distributed data warehouses for sale decision and it also present a solution for data transmission from local data warehouses to global data warehouse, which was based on large-scale clothing enterprises. In this paper, global data warehouse and local data warehouse are referred to as platform data warehouse(PDW) and enterprise data warehouse(EDW). PDW is built for e-commerce platform, and EDWs are built for the enterprise users who registered on the platform

There are two strategies for transmission between the PDW and EDWs. a). Round-robin scheduling, PDW extracts data from the EDWs; b). Data-driven, after EDWs completing update they transmit data to PDW immediately. Both of them are related to data exchange across database servers. The strategy a) is characterized by PDW based on specific conditions to determine the priority of the EDWs. Paper [5] presented using round-robin scheduling strategy in distributed data warehouses to solve the problem which is in the poor flexibility and real-time, but the global data warehouse had to maintain the additional views for update and the communication frequency was increased. The research also includes the relevant papers [6], [7]. Strategy b) is featured in EDWs extracting data in parallel then push them to PDW, and it has strong continuity, low network communication frequency and high concurrency. The adverse factor is that the strategy is very likely to cause conflict between the update transactions, and paper [8] proposed solution to the conflict from a data storage perspective.

## SYSTEM STRUCTURE

Business analysis system is aimed at building data warehouses to store the decision data,

 

analyzing the historical data and displaying readable results to the users. Based on the above objectives, the system structure as showed in Figure 1.

Business analysis system is classified as the presentation layer, application layer, and data storage management layer. Users can send data requests, then web server gets the request and OLAP server uses MDX query statement to

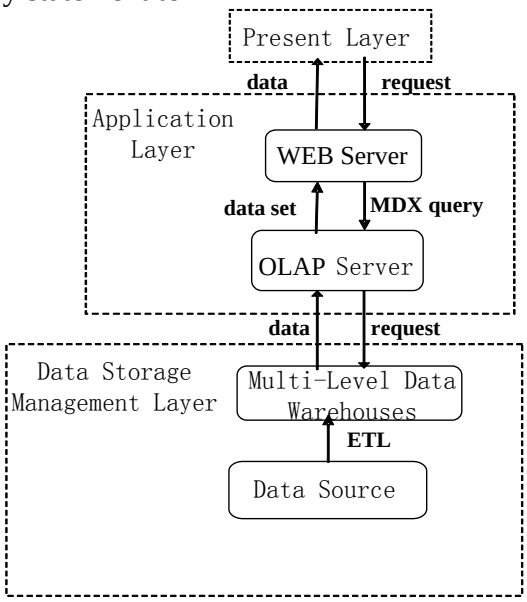


Figure 1. Structure of marketing analysis system

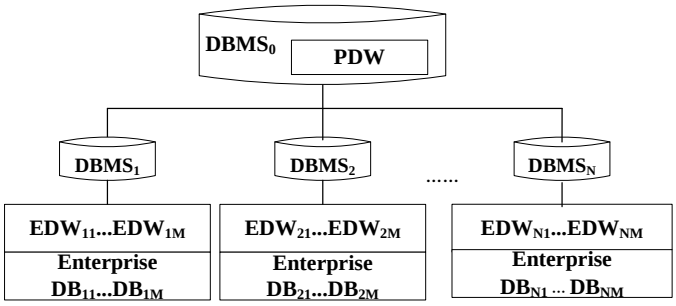


Figure 2. Data storage management layer

interacts with the data warehouses eventually result data is returned and displayed in the front page step by step. The open source tools can be used in the presentation layer and application layer. So this research focuses on data storage management—multi-level data warehouse construction. The platform is based on multiple database environment and multiple distribution databases up to homogenous in multi-database management systems. To achieve platform and enterprise-level different analytical functions, data storage management structure is shown in Figure 2.

Tablespaces are allocated for different users whose business models are independent. And the same operating mode determines corresponding to the same table structure in the separate tablespaces. All the databases and data warehouses used Oracle 11g database management systems. The data sources of PDW are the statistic from the registered EDWs. And data sources of the EDWs are taken from the corresponding database. Demand for analysis and decision making for different points of view, both logical data model and data granularity for the two level data warehouses also varies.

## LOGIC MODEL DESIGN

The main purpose of analysis can be described as the follow:

- Make users master the product marketing of their enterprises and propose the new marketing strategy.
- Tell users what merchandise sells well by analyzing the whole registered enterprises' marketing.

From the above, it can be deduced that enterprise-level analysis focuses on specific commodity marketing and platform-level analysis emphasizes on the categories of product marketing. Thus, the analysis granularities for the PDW and EDWs are different. In order to facilitate the subsequent statistics, we added the statistical values as a column to the fact table which is in the PDW. Based on a star model and snowflake model features and advantages, logic models for sale topic are designed as the Figure 3 and Figure 4.

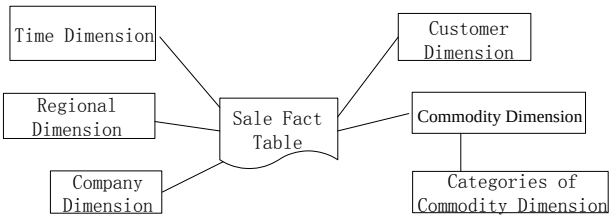


Figure 3. Logic model in a EDW

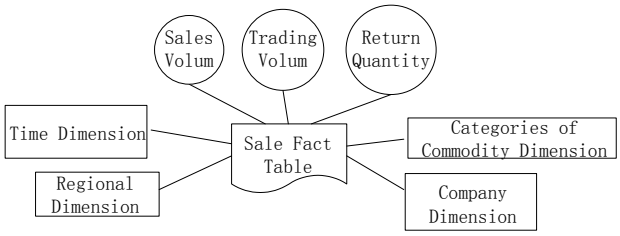


Figure 4. Logic model in PDW

Centered on sales fact table and creating the dimension tables about time, region, merchandise(or categories of merchandise). Especially, design for sale fact table can be expressed as that:

$$SALE\_FACT=\{A, P(ai)\} \quad (1)$$

$$A=\{a1,a2,..., an\}, P(x) =\{count(x), sum(x)\} \quad (2)$$

"A" represents the set of attribute values as the columns in the fact table, and the ai stand for a specific attribute value; P(x) is operational set which include COUNT and SUM as the grouping function in database, which is used to calculate the orders for sale, return order and sales amount. Attributes to this design, amount of data transferred between the single EDW and PDW in one day will be limited to N tuples. The integer N depends on the count of category dimension table.

## DATA PROCESSING

Due to use the same database management system and have identical table structure between the multiple database, the data sources are non-heterogeneous and not having dirty data. Based on the data model design, dimension tables and the fact table can be categorized by the update frequency.

- The tables which almost do not need to be updated: the time dimension tables and regional dimension tables.
- Periodic update of tables: the commodity dimension tables, customers dimension tables and companies dimension tables.
- The fact tables, which are updated every day.

The trigger can be utilized to the periodic update of tables. And updating strategy for the fact tables should be mainly designed. Each EDW counts its marketing results. Then the EDWs which complete the statistic transmit the data set to PDW. The ETL process for data transmission from enterprise databases to EDWs, or from EDWs to PDW will mainly utilize the operations: Mapping($\prod$), choice($\sigma$) and join( ).So they can be implemented by encapsulating as a procedure.

And there are two transmission strategies summarized in this paper— round-robin scheduling and data-driven.

### *A. Round-Robin scheduling*

PDW extracts data from the EDWs according to some algorithm. The main difficulty of the strategy is selecting the appropriate scheduling algorithm which must ensure that minimizing the time overhead and network traffic between the two level data warehouse, and reducing the maintenance cost. Classic genetic algorithm is superior to the traditional search algorithm to get the approximate optimal solution sets, but it's cumbersome, difficult to achieve and not conducive to maintenance. Paper [7] proposed the task scheduling based on a greedy algorithm, but it had much prerequisite and some of them can't be met in this case. The strategy in paper [5] is according to the system resource consumption adjust their algorithms in real time. But it didn't give the assignment priority rule. Serial extraction from EDWs by company code does not need to consider the complex scheduling algorithm. It's easy to implement and do not need to create the tables to record the communication data and scheduling rules. But it has to modify the procedure when a new company registering on the platform. Considering the flexibility not suitable for practical application, we chose the data-driven as the way to transmit.

### *B. Data-Driven*

After EDWs completing update they transmit data to PDW immediately. The time overhead T is showed in the following:

$$T=\max\{(ta1+tb1'),(ta2+tb2'),...,(tan+ tbn')\}+\Delta t \quad (3)$$

The tai represents time overhead of one EDW extract data from the corresponding database; And the tbi stands for the transmission time from a EDW to PDW; $\Delta t$ is extra time overhead. As the equation, the goal is to minimize extra time overhead. An experiment showed that the extra time was caused by the conflicts between multiple tasks, the segment in database automatic expanding and multiple network connections. Paper [8] proposed that utilize ORACLE's functionality on distribution to solve the conflict problem. Technical documentation [9] also provides solutions to reduce network connections. The specific approach is using list partitioning technology[10] to partition the fact table in PDW according to the value of company code. Next, the paper will verify the method's validity by the simulation experiment

## EXPERIMENTAL ANALYSIS

Efficiency of two-tier data warehouse update was modeled on the actual platform environment simulation test. Servers hardware configuration parameters as showed in Table I.

TABLE I. SERVERS HARDWARE CONFIGURATION

| | **Configuration Parameters** |
|---|---|
| Routing Server | Intel(R) Xeon(R) 8 CPUs 2.0GHz,6.0G RAM |
| Database Server 1 | Intel(R) Xeon(R) 4 CPUs 2.0GHz,4.0G RAM |
| Database Server 2 | Intel(R) Xeon(R) 2 CPUs 1.6GHz,5.0G RAM |

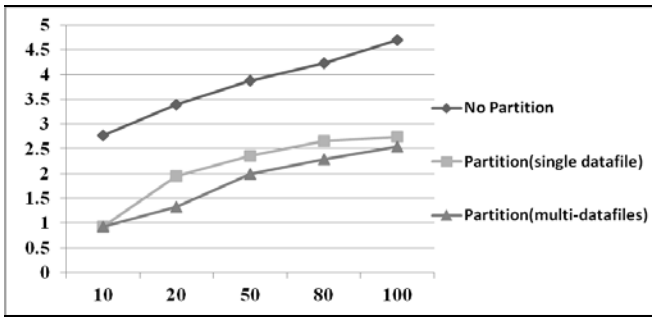


Figure 5. Time overhead of three cases

Establish data warehouses for enterprise users separately in the two database servers; Construct data warehouse for the platform in the routing server. Set time to ensure the system time of three servers consistent (the error is within 1 second). In order to close to the practical application environment, choose the time overhead which was in the nighttime and repeated to measure average as experimental results. According to the established design, compare the time overhead in the three cases:

- No partition fact table.
- Partition in a single datafile
- Partition in multiple datafiles

Experimental result is shown at the Figure 5. The amount of EDWs and time consumption(unit: s) is described in the X axis and Y axis respectively. Experiments show that under the same conditions, time consumption of transmission from the enterprise databases to EDWs and then to PDW can be reduced through table partition technology. Further analysis found that, the $\Delta t$ is mainly caused by the time difference of job delay in the two DBMSs which were established in the two different database servers. Table partition can reduce conflict between the multiple inserting tasks and then reduce the time difference, which may attribute to the storage features of the database. And experiments show that this method does not cause any data loss and alteration.

This paper proposes that limit the number of data by transmitting the statistics from EDW to PDW; utilize the data-driven to ensure high flexibility and consistency of the transmission; the table partition can be used to reduce time consumption in this application environment.

## Acknowledgement

This paper funded by : Project of Construction of the central financial support for Universities and Colleges (Project No.：P X M2014_014212_000097); Beijing Higher Education Young Elite Teacher Project(Project No.: YETP1419 ); The Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality (Project No.: IDHT20130502); Beijing Higher Education Teachers Team Construction Special Training Project---2014 General Abroad Visiting Program for Young Backbone Teachers ( Project No.: 067145301400)

## References

[1] Mehdi Kashfi, Abdolreza Hajmoosaei. Optimal Distributed Data Warehouse System Architecture: Big Data and Cloud Computing, 2014[C]. Sydney: IEEE, 2014:110-115

[2] Li Weiwei, Li Mei, Zhang Yang etc. Research of Classification Analysis for Distributed Data Warehouse[J]. Application Research of Computers, 2013, 30(10):2936-2939,2043

[3] Xiong Zhongyang, Huang Hailong, Zhang Yufang etc. Multi-level Data Warehouse Architecture and Update Algorithm of Double Channels View [J]. Computer Science,2002, 29(6):79-81

[4] Ye Zheng. Design of Disributed Data Warehouse for Sales Decision of Large-scale Clothing Enterprise [D]. Zhejiang: Zhejiang University, 2006

[5] Yang Yiping. Research and Design for Data Scheduling Based on Distributed Data Warehouse [D]. Beijing: Beijing University of Posts and Telecommunications, 2006

[6] Wang Shan, Chen Kun. Research on Task Scheduling Method Based on the Greedy Algorithm in ETL [J]. Microelectronics & Computer, 2009, 26(7):130-133

[7] Zhong Qiuxi, Xie Tao, Chen Huowang. Task Matching and Scheduling by Using Genetic Algorithms [J]. Journal of Computer Research and Development, 2000, 37(10):1197-1203

[8] Liu Peiyu. A Study on Sharing Techniques for Multi-users in LAN [J]. Computer Engineering and Applications, 2000(07):114-115,141

[9] Steve Fogel, Tony Morales, Padmaja Potineni, Sheila Moore. Oracle Database Administrator's Guide[EB/OL]. (2008-03)[2015-01-06] http://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts002.htm#ADMIN12085

[10] Muxiu Zhu, Xianjiu Zhang. The Management of Big Data Tables Based on Oracle Partition Technology: Computer Science and Education, 2014[C]. Vancouver:IEEE, 2014:570-572