Semantic Caching for Querying Data Warehouses in Applications

Yao Shuchun

Suzhou Industrial Park Institute of Services Outsourcing
Suzhou Jiangsu China 215123

Key words: Semantic Cache, Query Processing, Data Warehouse, Response Time

Abstract. This document presents three-tier architecture for data warehousebased Online AnalysisSystems in some applications. To improve the performance, semantic caching is introduced and implementedin the middle tier. The query results are cached together with semantic descriptions such that that next time similar queries can be completely or partially satisfied from the cache instead of the data warehouse. Thus, the communications with the data warehouse are significantly reduced. As a result, the overall response time is improved.

1. Introduction

This document talks about the central data warehouse based on Teradata Database(DB) System for an Information Management System, and the Business intelligence layer of this system based on the data warehouse infrastructure, which provide the company's business staff entrance for the application of data warehouse system.In order to meet the needs of large-scale application, we choose the Web based three layer structures to achieve the functions of statements and dynamic query.In order to provide dynamic content through web pages, the system must connect to the database to obtain the required information, so the performance of the system is affected by the number of connections in the database.If there are a large number of users and a large number of requests to access the DB at the same time, the DB query processing will become a bottleneck affecting the performance of the system.To improve the system performance, a better approach is to reduce the workload of DBduring many querying.So we propose semantic caching technology in this paper,to improve the performance of the system.If a client's request can be resolved through the data in its cache, it does not cause the network action, the client and server side of the query performance will be improved much.

2. AGeneral Overview of Semantic Cache

2.1 Three layer structure based on WEB

The three layer structure consists of the Presentation Layer, the Middle Layer and the Database(DB)Management Layer, and the relationship between the 3 layers is shown in Figure 1

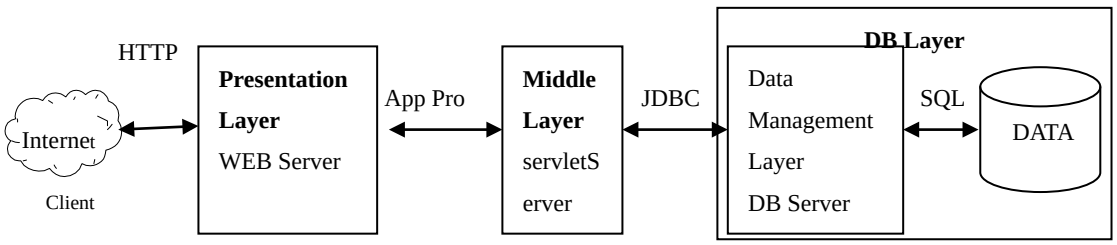

Fig 1.Schematic diagram of the three layers structure

Presentation Layer: to accept the user request, and pass the customer requestto the middle layer, and return the results to the middle layer user.

Middle Layer: responsible for the application of the logic of the implementation and to send a request to DB, and is responsible for the conversion of the database to return the data records and the results of the implementation of feedback to the presentation layer.

Database Management Layer: Receiving from the middle layer of DB query. Responsible for receiving the query request and the results returned to the middle layer data record. Be responsible for using the SQL statement to obtain the result of the target data from the database.

Three layers client application is more flexible for large capacity. The middle layer can handle different interface to the database, so the application can transparently access different data sources without affecting the client.In addition, when a large number of client requests, the middle layer is easy to expand, it can be allocated more server application execution.

2.2 Semantic Cache

In order to improve the efficiency, we introduce and implement semantic cache in the middle layer.Semantic caching can reduce the communication between the middle layer and data management layer, and the workload can be concentrated in the middle layer.When the middle layer of the heavy load, it can increase the ratio of processing server, database management layer expansion easier, because of the different database concurrency issues.In this system, the user's query results are storage in the middle layer of the semantic cache, if a client of similar requests can be resolved through the cache of all or part of the data, completely eliminated or reduced database access operation, the overall query performance will be has been greatly enhanced.The logic diagram is shown in Figure 2:

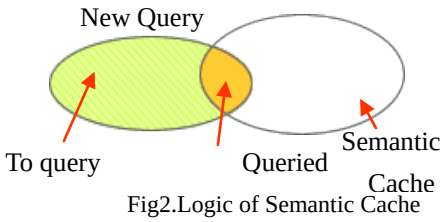

Fig2.Logic of Semantic Cache

The semantic cache will be regarded as a set of semantic regions, each semantic region aggregation semantically related data packets, such as a user's query request.Therefore, information access and cache rewriting are managed as a unit in the semantic area.After Using semantic cache, the new query is divided into two parts (as shown in Figure 2) when a query request is submitted by the client.Queried means that it can directly obtain query results from local cache(also called Semantic Cache), the remaining query refers to the need to obtain query results from the server database.If the local cache cannot provide all the query results of the new query, the system rewrite the new query statement, the remaining query part of the database server to submit the corresponding query results.Semantic cache shown as in Figure 3, there are 4 types:

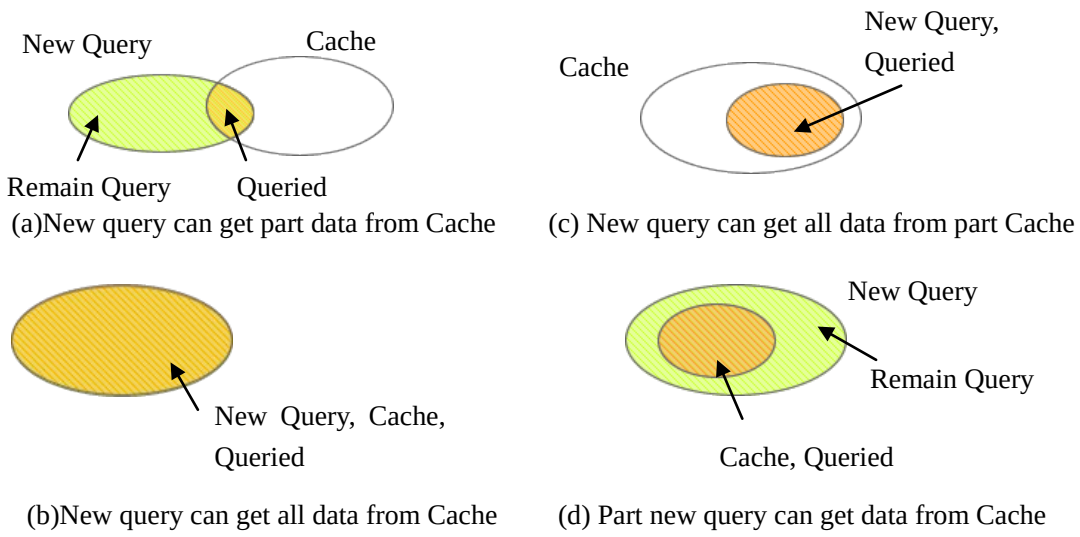

Fig.3.Four types of Semantic Cache

Type(a), the new query by the two part of the synthesis, has been found in the query part can be directly obtained from the cache results, the remainder of the query will be submitted to the database to obtain results.

Type(b), the new query and cache completely consistent, has been checked for the new query, the remaining query is empty, do not have to conduct additional database access.

Type(c), the new query is a subset of the cache, the remaining query is empty. Query results from local cache.

Type(d), cache is a subset of the new query that has been checked for the entire cache. New query from the local cache can obtain partial results, the new query by the system to re-describe, after the separation of the remaining query from the database to obtain the rest of the results of the query.

Semantic cache has many advantages such as low cache overhead, reducing network congestion, no need to communicate with the server to obtain the results of the query and so on.But in the semantic cache, the separation of the remaining queries is more complex, and the system cost increases with the increase of the attributes.

3. Semantic Cache Design and Implementation

3.1 Semantic cache application related system flow chart (see dashed line box):

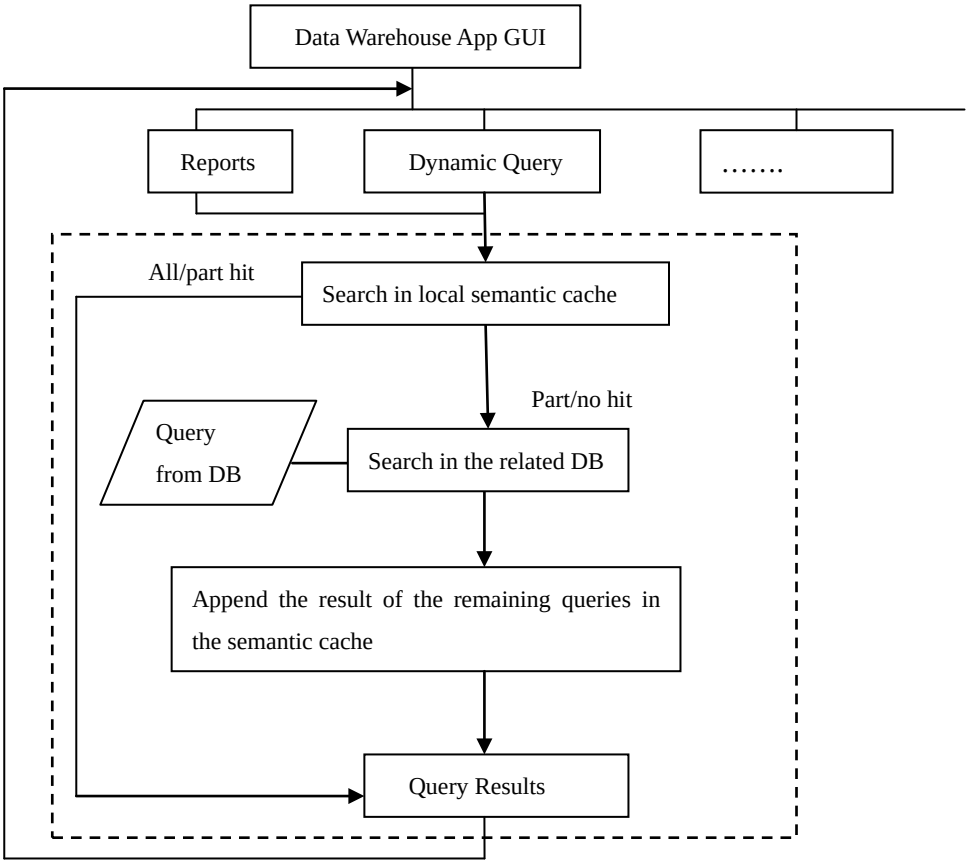

Fig. 4 Applicationsystem of semantic caching

In the flow chart, to implement semantic caching of the system includes several key functions:

1) To create a single semantic cache for each session when the user logs into a data warehouse application system.

2) According to the semantic description of customer queries, they divided into two parts have been checked and the remaining query, in order to obtain the results from the semantic cache.

3) Check the semantic cache, and give the result of the query.

4) Through the relevant DB to achieve the goal of the remaining query record, add new records and update the semantic cache, and give the results of the query.

3.2 semantic cache data structure

In order to identify and compare the semantic cache record, a user's inquiry was first re-described and stored in the cache, as well as from a database query result records data will also remain in the cache.Here, the semantic cache of the system is constructed by LinkedHashMap, which is a data structure of JAVA design, which is useful for the implementation of cache LRU "least recently used" replacement principle.LinkedHashMap is a hash table and link to achieve the image interfaceand you can insert the order of the key/value added to the link in the hash image.LinkedHashMap is also a HashMap, but the internal maintenance of a two-way linked list, basically and almost HashMap.HashMap is a hash table with image interface, which is composed of a set of keywords, values, and keywords/values. Hash table are similar to the list consisting of arrays, allowing users to easily insert and find information, the only difference is that HashMap allows the value of space, but also is not synchronized.When an empty link is created, the default value for the capacity and the load factor is 0.25 and 0.75.The size of the semantic cache size should not be too large, and otherwise, it will take more time in the search cache. Load factor is a measure of the hash table in its capacity to automatically increase before the index was taken

out.When the LinkedHashMap table mapping has been filled, the least recently used mappings will be removed, such as in the first mapping will be replaced, in the remaining query access to the records of the results of and inserted into at the end of the linked list of images. To construct the semantic cache LinkedHashMap structure as shown in figure 5:
LinkedHashMapThere is a key and a value mapping in LinkedHashMap. The keyword is used to identify each key value, in part by the hash table, the hash table storage parameters database query and query.Value section is used to store the result data record of the query.

In Figure 5, the keyword hash table by the query and query parameters, such as the composition of the hash table contains 3 entries: data tables, parameters and query. The data table entry is a vector that stores the table names of all the data tables involved in the query statement.The parameter entry is also a vector that stores all the parameters involved in the query statement.More than two entries will be used in the process of finding semantic cache.Query entry contains the integrity of the query statement submitted by the client, but the semantic cache record expires, the query statement will be submitted to the database to obtain the latest data records update cache.

The value portion of the LinkedHashMap is the Row Set Cached (cache line set), which is the object of the serial disconnection.In this system, the DB query result set is stored, which is a data table.This data sheet is usually generated by executing a database query statement.A pointer to the current row of data is maintained in a data table, which can be repeated in the result set.Cache line set (cached row set) is a Java provides a function, through the interface of the javax.sql.RowSet provide to connect to the database, to the result set in the form of access to data, release the connection and function in the local cache operation data.

In addition to the LinkedHashMap structure, the system also enables another Hash table, which is used to save the last modification time for each record in the cache.The key part of the LinkedHashMap and the structure of the same key parts of the structure, the database query and query parameters.The value part is a time mark, when a cache record is generated, it is used to identify the record generation time.When the cache record is replaced or modified, the corresponding time flag is also updated.

3.3 Implementation of Semantic Cache

Semantic cache is realized in the system similar to the DB cache, including the realization of

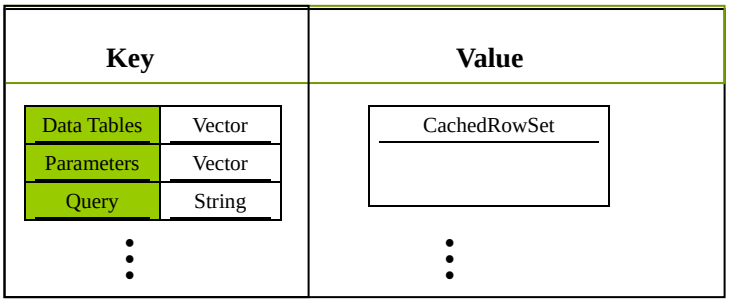

Fig.5 Constructing Semantic Cache LinkedHashMap

semantic caching, establish semantic cache retrieval, semantic cache update etc.In the following example, semantic cache retrieval is a detailed description of the implementation process.

In the analysis system, when a user login system, the semantic cache is created and stored in a HTTP dialog.So each active user of the system will be corresponding to a single semantic cache.The cache after the establishment of every customer's request by the system to describe, exist in the corresponding hash table.The corresponding query results are also stored in the semantic cache LinkedHashMap, and the results are expressed as feedback to the user.In addition, the last

modification time of the cache record is inserted as a new entry into the Hash table.The semantic cache will be released when the user exits the system or the process dialog is timed out.

When a user submits a detailed DB query request, the system searches for the semantic cache in the first place and The data flow chart of semantic cache retrieval, as shown in Figure 6.This new query is re-described and store in a hash table, and then compared with the keyword in the LinkedHashMap, the comparison is carried out by the function containsValue().If fully the same as a key in the cache, and the corresponding record is still valid, the query results returned from the cache access.If it have over time, according to the relevant DB to refresh the relevant content, so that the cache will continue to be updated in a timely manner.

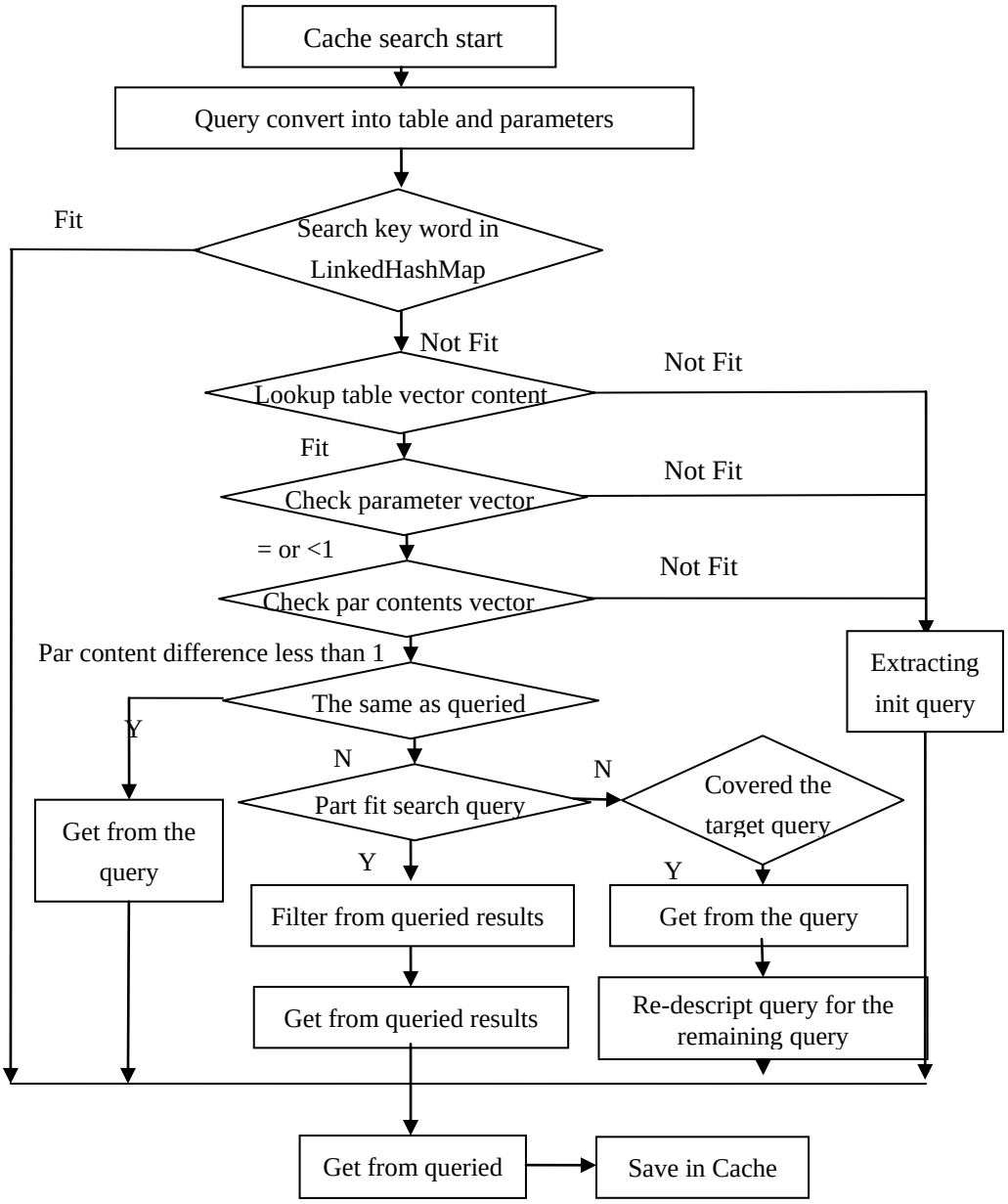

Fig. 6 Data flow chart of semantic cache retrieval

If the new query and LinkedHashMap in the key word is different, then continue to search keywords in order to find the individual entries.The re-described query is compared with the data table and the parameter vector in the hash table mapping. If the vector is exactly the same, the result set of the valid cache can be directly returned to the user.However, you will meet with all the key

words in the LinkedHashMap query. So we need to consider the situation of matching part of the key words.Here, suppose that the matching part of the key word is the same as the entry of the new query and the hash table, and there is only one difference between the two. This difference may be the number of parameters, or the same number of parameters and values are different.Select a different point for the part of the search for a complex problem of simple considerations.Otherwise, the multiple points will have a very complex combination.In addition, because the semantic cache returns the first part of the search process that is encountered in the search process, it may not be the best part of the query to match the query.

According to the flow chart shown in Figure 6, the data table name, the size of the parameter vector, and the individual parameters are compared in turn, if there is a difference between the two, the semantic caching and query.If the new query is part of the hit cache, the new query is broken down into two parts, the search query and the remaining query.

There are many kinds of coverage for query and semantic cache, such as, the results of the query has been found to contain exactly, partly contains and really contains the results of the input query.When the query and input query have the same parameters, the results of the query can be satisfied with the input query, you can directly return the results of the query.If check query and the input query is partially covered, the input query to form a remainder query and database communication to get the remaining part of the query results, will semantic caching and database query result from a combination of whether to return results.This input query is also inserted as a new entry in the semantic cache.In this way, the same query can be directly obtained from the cache query resultsnext time.A sample graph of some hit queries, as shown in Figure 7, is a combination of the results from the semantic cache and the database query, as well as a number of parameters.

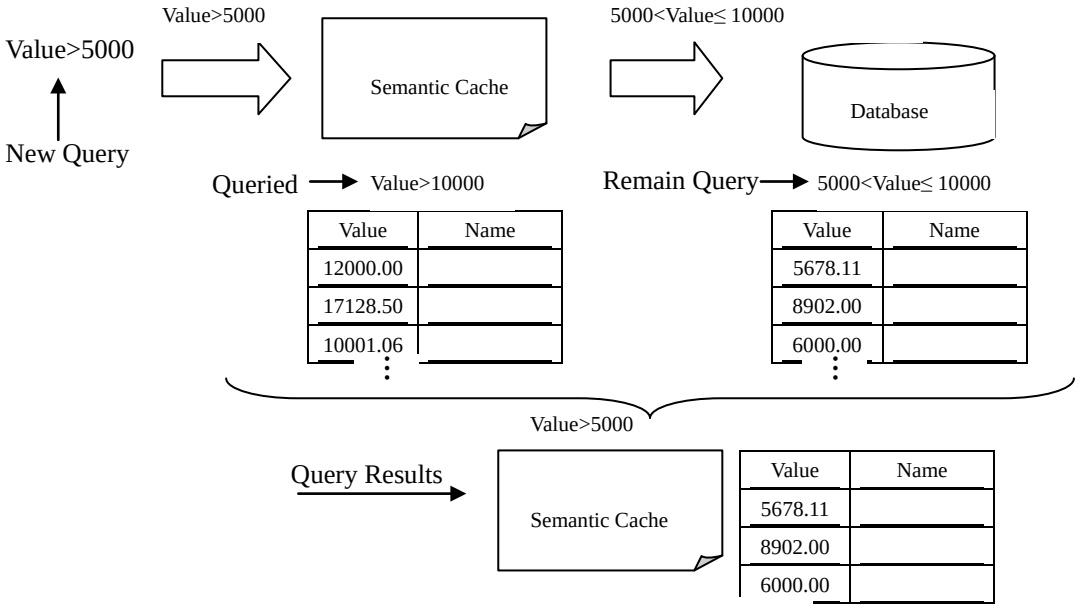

Fig. 7 Fit queryexample

When a query has been found that contains the input query that has been found to check the results of the query set is greater than the input query result set, according to the requirements of the query filter has been found in the results of the query record, form the results of the query returns (no longer for example here).At the same time, the query result set is inserted into the semantic cache as a new entry.

A brief exposition of the system's semantic cache,when the semantic cache belongs to the user to update the relevant database data, the user's semantic cache will be updated.Users to update

instruction is first sent to the database system and the update instruction after the transformation, finding semantic cache data table and parameter, all by the influence of entries from the database update data, so as to ensure cache contains is the new record.If other users update the database, the users' semantic cache records only then cache the record over time to trigger the update operation.Because of the different data table records update with different frequency, time period settings will be different.

3.4 Semantic Cache Performance and Limitations

3.4.1 Semantic Cache Performance Performance

Actual operation results show that application with semantic caching have fast response than traditional, in the realization of the function of simple and routine showed better performance, for example take few and simple dynamic query etc.But for the advanced and complex query analysis applications, the response time of semantic caching system will be extended, but the average response faster than the traditional system.This is because of the features such as simple query on semantic cache hit rate is higher, and the query results can be obtained without direct access and database directly from the cache.In other functions, the semantic caching system is similar or longer to the traditional response time.Advanced analysis results are usually obtained from a random non repeated query combination, which is low in the semantic cache, which may be a part of the hit or miss.Also such a query request rarely repeat or similar to previous queries, often need to search the entire semantic caching, and must have access to the database query results, so the response time of semantic caching becomes longer.

3.4.2Semantic CachingLimitations

The update time of the semantic caching system is longer than that of the traditional DB.When the DB is updated, the semantic cache must be updated.System has to be found in the cache response to the table to update, and it needs more time to update records.When the number of the need to update the data table is increased, the time required will increase, because the system had to search all the entries in the cache, according to the contents of the database to refresh the query results.

The semantic cache update of the system uses the LRU method, so only the latest query will be saved, and those are not used frequently will be replaced.The semantic cache needs to be continuously updated, and the data can be retrieved from the corresponding DB when the effective time of the cache is expired.Only for the individual cache update cache, so when other users updatedDB, the individual cache storage may benot the latest DB query results. Because of their dependence on time validity judgment the cached records of database synchronous system, the design of prescription is more particular about the may be different queries with different aging time, and the failure time of the system and the database table is accessed frequency and preference for using have greater relevance.

4. Conclusions

This article in the information access based on Web mode of three layers of structure of a data warehouse application analysis system, introduce and realize the semantic cache and reduce the communication between the middle layer and data management layer and improve the user request response. This article introduces the concept of semantic cache, and its design idea and realization method, and performance.

References

[1] *XU Fangfang,LIYaoyao,GUJinguang.* Semantic Cache Replacement Strategy for XML Algebra-Based Query Optimization[J]. Wuhan University Journal of Natural Sciences,2015,v.20;No.10002:165-172.

[2] Answering queries using cooperative semantic cache in mobile computing environments[J]. The Journal of China Universities of Posts and Telecommunications,2012,v.1903:54-59.

[3] *K. SelçukCandan, Wen-Syan Li, QiongLuo, Wang-Pin Hsiung, DivyakantAgrawal,* Enabling Dynamic Content Caching for Database-driven Web sites, ACM Press, 2001.

[4] *Boris Chidlovskii, Claudia Roncancio and Marie-Luise Schneider,* Semantic Cache Mechanism for Heterogeneous Web Querying, Computer Networks, 1999.

[5] *Dongwon Lee, Wesley W. Chu,* Conjunctive Point Predicate-based Semantic Caching for Web Databases , 1998.

[6] *Dongwon Lee, Wesely W. Chu,*Semantic Caching via Query Matching for Web Sources, ACM Press, 1999.