Journal of Information and Operations Management
ISSN: 0976–7754 & E-ISSN: 0976–7762 , Volume 3, Issue 1, 2012, pp-280-283
Available online at http://www.bioinfo.in/contents.php?id=55

TECHNIQUE FOR QUICK ACCESS FROM DATA WAREHOUSE

RITIKA AND RAKESH KUMAR SAINI

Department of MCA, Dehradun Institute of Technology, Dehradun (Uttrakhand), India.
*Corresponding Author: [1]riti_79@rediffmail.com, [2]rakeshcool2008@gmail.com

Received: December 12, 2011; Accepted: January 15, 2012

Abstract- Data warehousing is a buzzword in the database industry where data is stored for archival, analysis and security purposes. Database research community attention is on store house where large amount of data is stored which make easy to access data and use it in day to day operations. In this paper we will try to find the common problems faced. These kinds of problems are faced by data warehouse administrators and users. Here we outline some query performance techniques which minimizes response time and improves overall efficiency of data warehouse, particularly when data warehouse is access and updated frequently. By and large performance of the system is improved without accessing the original information sources which provide good strategies that make finer data warehouse.
Keywords- Integrated Data Warehouses, Operational database, Data Access tools Response time, performance.

Citation: Ritika and Rakesh Kumar Saini (2012) Technique for Quick Access from Data Warehouse. Journal of Information and Operations Management ISSN: 0976–7754 & E-ISSN: 0976–7762, Volume 3, Issue 1, pp-280-283.

Copyright: Copyright©2012 Ritika and Rakesh Kumar Saini. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Introduction

A data warehouse is a collection of subject-oriented, integrated, nonvolatile, and time varying data to support decisions makers. A data warehouse is the copy of transaction data especially structured for querying, reporting and analysis purpose. Analysts and Business users use the data warehouse to answer unlimited variety of questions, which may be very difficult to answer in operational database. Data are acquired from various operational data stores across the enterprise. After acquisition of the data, cleansing, transformation and possibly denormalization is been applied for better performance. The primary goal is creating data access methods which are very helpful for quickly access data from data warehouse. Data warehouses are large repositories of data that contain historical and integrated data collected from a number of heterogeneous sources. Query performance is a major issue as most of the queries are complex and adhoc in nature and require huge volumes of data to be processed. Various techniques have been proposed by different researchers to improve the query performance in a data warehouse environment. Here we have discussed few techniques that are used in traditional database systems to boost query performance.

Query Performance Techniques

Performance issues in data warehousing are centralized around access performance for running queries and incremental loading of snapshot changes from the source systems. The following techniques are considered for a better query performance:

Balance components for data warehouse

When configuring a Data Warehouse system it is important to insure that all components are balanced. What this means is that all active components (CPU, Memory and I/O) are all effectively used and when one component reaches its maximum operating range, all the other components approach their maximum at the same time. This practice has been perfected by engineers for centuries [6]. The strength of a structure is determined by the weakest link in the structure. In addition an engineer will optimize the structure to use the most cost effective materials and construction methods to build the structure. This principal can be applied to configuration of the data warehouse. We know that the CPU is the most expensive non renewable resource on the system. With this fact appreciated the memory and I/O subsystem should be constructed to ensure that the CPU can be kept busy and the design held in balance. Failure to do this makes the memory or I/O subsystem the weak

link in the structure.

CPU requirements for data warehouse

CPU sizing and requirements for a Data Warehouse are a function of the database size and the user workload. The database size will impact the number of CPUs simply because a database needs to be administered [5]. The database requires periodic reorganization which requires extracts of data, data reloads and data re-indexing. A database requires backing up and restoring in the case of a media component or operational failure. All of these processes require CPU resources. As the data volume increases it becomes more important to run these operations in parallel to make the database batch windows and availability requirements [8]. For this reason the database size will have a large impact on the number and size of the CPUs required. As a rule of thumb if the database size drives the CPU requirements then recommended would be to assume about 5 Giga of data per CPU, assuming fairly conventional availability requirements. This rule is not a true linear rule because when the database size increases some economies of scale can be exploited. Also as systems increase in size their I/O subsystems tend be upgraded and more efficient. When adding additional CPUs to a system the scalability with regards to additional CPUs should be considered and fully understood [3]. It is naive to assume in a multiple CPU environment that you will achieve linear scalability on all operations for all the CPUs. It is important to understand which operations scale well and under what conditions which operations are inherently single threaded and do not scale at all. In summary characteristics of CPUs are simple and predictable. The increase in throughput is a function of the scalability of the system.

Use best query techniques for data warehouse

One of the most important query types that will be exploited within a Data Warehouse practice is the star query. As great deal of Data Warehouses are constructed using a star schema. To fully exploit this model the Oracle kernel from 7.2 onwards will attempt to optimize star queries [4]. A star schema consists of a central intersection table or more commonly called “fact” table with a large number of satellites tables related to the central table. The satellite tables define the dimensions of the star query. An example of a star schema would be a central sales table that stores the value of each sale. On the fact table there would be a large number of foreign key references to the satellite tables. These could be product, product type, business period, geographical region, and sales organization. The star query is composed by referencing the outer satellite queries within the query ‘where’ clause predicates. The specification of all the join conditions insures all the tables are correctly linked.

(iv) *Extraction, Transformation and Loading (ETL)*

The information contained in a warehouse flows from the same operational systems that could not be directly used to provide strategic information which makes a difference between the operation system and Information contained in the data warehouse. It is the set of functions that fall under the broad group of data extraction, transformation, and loading (ETL).

- **Data Extraction – The data contained in the**

Data warehouse is extracted from the organization operational store as shown in Figure 1. The data in the data warehouse can also be obtained from various external sources. Sources such as flat files, html, xml documents, text etc.

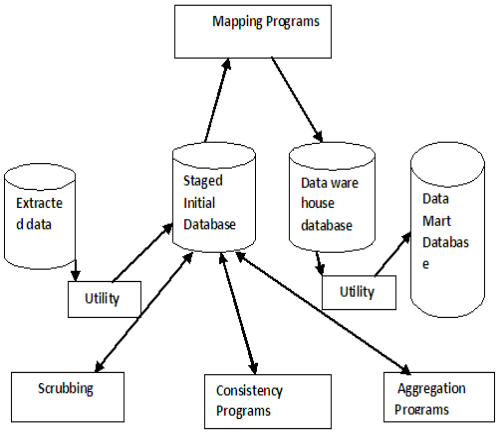

Fig. 1- Transformation and Loading

- **Data Transformation–** Data transformation Techniques have been applied on the data to make it more uniform. After transformation the resulted data is more homogeneous and have less inconsistencies and errors [2]. This enhanced the performance of data warehouse.
- **Data Loading –** After the cleansing and Transformation of the data, uniformed data is placed in the warehouse. The above tasks are often referred to as ETL –extraction, transformation, loading as shown in Figure1. They are frequently not well-defined distinct phases as suggested above. The extracted data will often be stored in a central staging area where it will cleanse and otherwise transformed before loading into the warehouse. An alternative approach to information integration is that of mediation: data is extracted from original data sources on demand when a query is posed, with transformation to produce a query result.

Optimization of data mart

Data marts are a critical part of data warehouse design. They are created to store data required for analysis by specific departments e.g. manufacturing department, sales department or inventory control.

Data is extracted from the warehouse and transformed into facts and dimensions within a data mart, which is then queried using client tools or business intelligence reporting tools. Information builders offer comprehensive solutions for developing, managing, and optimizing a flexible data architecture that efficiently supports any enterprise information initiative [12]. Our data management, ETL, and real-time transformation solutions meet the widest range of information needs by enabling direct access to data in operational systems, data warehouses, and data marts, or retrieval of detail data via drill-throughs.there are used different access techniques to produce high-quality data, a reusable infrastructure, and comprehensive metadata, including:

- Using ETL to extract, transform, and load data directly into a data warehouse or data mart.
- Directly accessing operational data stores or the files that service operational systems
- Transparently drilling and joining data warehouses to operational data.

- Using an operational system's own application functions to access data.
- Trickle-feeding a data warehouse to populate and refresh it.
- Performing sophisticated data integration, transformation, and manipulation in real-time.

Data management strategy

The data management strategy is comprised of disciplines involved with specific segments of data management and these segments are governed by standards and best practices. Structure and design focuses primarily on data modeling and artifacts that flow from data modeling activities. The goal is to provide direction on modeling practices, the use of conceptual, logical, and physical models, naming standards, and modeling tools. Data storage focuses on the management of persistent data stores throughout the organization. It covers such topics as database selection guidelines, backup & recovery procedures, data retention/archiving policies, and disaster recovery planning [9].

Data movement focuses on the movement of data between systems. In this context, "systems" include external data sources, operational systems, and analytic data stores.

Use Snowflake data model schema

A snowflake Schema [11] in its simplest form is an arrangement of fact tables and dimension tables [8]. The fact table is usually at the center surrounded by the dimension table. Normally in a snow flake schema the dimension tables are further broken down into more dimension table. Snow flake schema is one of the types of designs present in database while the other one is the star schema [13]. It is of Normalized form. Star and Snowflake Data Models are typically used to support Data Warehouses or Data Marts.

These designs are characterized by a central Table, holding the 'FACTS', with a number of 'DIMENSIONS' tables radiating outwards, just to one level. The snowflake schema is an extension of the star schema, where each point of the star explodes into more points. In a star schema, each dimension is represented by a single dimensional table, whereas in a snowflake schema, that dimensional table is normalized into multiple lookup tables, each representing a level in the dimensional hierarchy.

The main advantage of the snowflake schema is the improvement in query performance due to minimized disk storage requirements and joining smaller lookup tables.

Use Query cache

- We can improve query performance by using the query cache. Query cache will store all record of executed query. Query cache will keep record of newly executed queries. The purpose of the query cache is to reduce the response time of query. It will increase the brainpower of data ware house so that system will memorize the latest work it has performed.[5] The query cache is extremely useful when your application executes a particular query again and again and tables don't change very often thereby avoiding the overhead of running through data over and over and thereby increase the execution time. In case your tables change very often or if your queries are textually different every time, the query cache may result in a slowdown instead of a performance improvement. Caching query results can dramatically improve script execution time and resource requirements. There is query cache provide features: Connect to the database
- Prepare query
- Send query to database
- Retrieve the results
- Close the database connection

If the query is stored, then check the state is valid or invalid. If state is valid then data can be access and if state is invalid then data can't be access. but If user send a query of insert, update, delete and drop then data will be alter in database and state of related query will be invalid. Now invalid state data and query can't be access by user. This can save important time and improve data warehouse performance by not reevaluating the queries which are already stored in the cache. One of analyst place a query to show me the products of a organization, Then this query will look like as follows: -

Query1: SELECT product_id, product_name, product_ price, supplier_id FROM produt_information WHERE supplier_id =501.

When the query is submitted, query cache will be examined to check whether this query is available or not and state is valid or invalid [7]. If it is not available, query will be evaluated and result will be store in the query cache [10]. The results of the query are shown in the Table1.

Table 1- Result Of Query1

Product_id	Product_name	Product_ price	Supplier_id
301	Motherboard	5000	501
302	CPU	7000	501
303	Monitor	4000	501
304	Processor	9000	501

Whenever any other user submitted the same query the result will be retrieved from query cache because that query is already stored in the cache. We will call this Query1. Suppose another user wants to retrieves the information of products whose product_price greater than equal to 7000 AND supplier_id is 501. Then the query will be as

Query2: SELECT product_id, product_name, product_ price, supplier_id FROM produt_information WHERE supplier_id =501 and product_ price>=7000.

When the query is submitted, cache memory is examined. Same query is stored in the cache memory and state is valid then we can get the result in Table 2.

Table 2- Result Of Query2

product_id	product_name	product_ price	supplier_id
302	cpu	7000	501
304	processor	9000	501

Now output of Query 2 will be generated from the Query 1 result set instead of going through from all the data stored in the data warehouse. This process will save lot of time and effort required to go through all the records. Queries against complex view definitions must be answered very fast because users engaged in decision support activities require fast and quick answers to their questions [7]. Even with sophisticated optimization and evaluation techniques, there is a limit to how fast one can answer such queries.

The main purpose of a query cache is to improve query performance.

A great deal of time in any query can be wasted by unnecessary sorting and filtering. Some of the most common of these may involve the use of a "union" statement where a "union all" would have been suitable. In this case the "union all" eliminates the need for a sort to eliminate duplicate rows in the set operation [2]. A more common example is when using an "order by" clause. If an index range scan is used and the index is built according to the ordering specified by the "order by" clause the query is able to retrieve the data presorted and then eliminate the sort operation to satisfy the "order by" clause.

By improve index definition

Poor index design is responsible for great number of performance problems. The selectivity of an index can be greatly increased by the use of concatenated index to include many of the "where" clause predicates and hence reduce the size of an index range scan and the subsequent number of table lookups [1]. Further Refinements can be made to the design of a concatenated index by appending the remainder columns from the select list of the query such than not table lookup are performed in the query at all. In this case the query is entirely satisfied from within the index. Another issue index designs is that very often little thought is given to key ordering when specifying a concatenated index. If possible is best to place the columns in order of selectivity with the most selective column first. This is particularly important to verify index definitions and designs generated by case tools as these tools tend to generate the index definition according to the Dictionary ordering of columns and not by selectivity of the columns.

Conclusion

In this paper some query performance techniques for speeding up queries in a data warehouse has been Proposed. These techniques improve the performance of data ware house by minimizing the response time significantly. Query Cache technique is to store queries and their corresponding results. If similar query is submitted by any other user the result will be obtained using cache memory.ETL functions needed to be carried out by a competent and trained ETL team. It's the responsibility of ETL architect to devise a comprehensive and effective ETL process to load the data warehouse. The ETL architect/expert ensures that the ETL processes have strength and endurance. All these techniques are used to provide fast answers to user queries. Snowflake Schema plays a key role in improving data warehouse performance and getting the desired out put from it. Optimization of Data Mart also plays very important role in improving data warehouse performance.

References

[1] Lingra, P.J., Yao Y.Y. (1998) *Journal of the American society for information scienct.*49 (5), 415-422.

[2] Albrecht J., Hümmer W., Lehner W., Schlesinger L. (2000) 4*th International Conference on Flexible Query Answering Systems*.

[3] Yu C.T., Sun W. (1989) *Automatic Knowledge Acquisition and Maintenance for Semantic Query Optimization, in: IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

[4] Shim J., Scheuermann P., Vingralek R. (1999) 11*th International Conference on Scientific and Statistical Database Management*.

[5] Kachur Richard (2000) *Data Warehouse Management Handbook", Prentice Hall*.

[6] Poe Vidette (1969) *Building a data warehouse for decision support", Prentice Hall*.

[7] Bischoff Joyce and Alexander Ted (1997) *Data warehouse, Practical advice from the experts, Prentice hall*.

[8] Dr. John. D Porter and J Rome john. *Arizona State University, "Lessons from a successful data warehouse implementation*.

[9] Silberschatx A., Korth H. Fand Sudarshan S. (2001) *Database System Concept, Mc Graw Hill*.

[10] Chan C.Y. and Ioannidis Y.E. (1999) *ACM SIGMOD*.

[11] Berson A., Smith S.J. (1977) *Data Warehousing and Data Mining, McGraw -Hill*.

[12] The TPC Benchmark (TPC-H) *Transaction Processing Performance Council*.

[13] Neil P.O., and Quass D. (1997) *ACM SIGMOD International Conference on Management of Data*, pp. 38-49.