Deadline and QoS Aware Data Warehouse

Wen-Syan Li,* Dengfeng Gao, Rafae Bhatti, Inderpal Narang
IBM Almaden Research Center

Hirofumi Matsuzawa, Masayuki Numao
IBM Tokyo Research Laboratory

Masahiro Ohkawa, Takeshi Fukuda
IBM Yamato Software Laboratory

ABSTRACT

A data warehouse infrastructure needs to support the requirement of (day time) ad hoc query response time and (night time) batch workload completion time. The following tasks need to be finished in a batch window: (1) Apply one day's delta data to the base tables; (2) refresh MQTs (Materialized Query Tables) for ad hoc queries and batch workloads; (3) run batch queries. Tools are available to optimize *each step*; however, many factors need to be considered for improving the overall performance of a data warehouse (i.e. meeting batch window deadline and ad hoc query response time). We have prototyped a *Data Warehouse Operation Advisor* to systematically study each component contributing to the batch window problem, and then perform global optimization to achieve desired results!

1. INTRODUCTION

Data Warehouses and Business Intelligence (BI) applications are the top two priorities for CIO/CTOs (Celent report on December 14, 2005). Many large enterprises have daily transactional information stored at the branch level computer systems. The information is transmitted to the head quarter data warehouse using ETL (Extract, Transform, and Load) tools in the early evening for consolidation and analysis. The information from various branches is integrated into a centralized enterprise data warehouse and complex BI applications are executed in the late evening. A materialized view or Materialized Query Table (MQT) is an auxiliary table with precomputed data that can be used to significantly improve the performance of a database query. MQTs are essential to BI application performance. A Materialized Query Table Advisor (MQTA), such as DB2 Design Advisor [1, 2, 3], is often used to recommend and create MQTs for enterprise data warehouse (EDW) operations.

The BI applications are run as a batch query workload on the EDW. In addition to the batch query workload in the evening, there are ad hoc queries issued in the daytime to find information that is not part of the batch query workload. A typical operation of an enterprise data warehouse has the following steps:

*Corresponding author. Email: wen@almaden.ibm.com

VLDB '07, September 23-28, 2007, Vienna, Austria.

1. The transactional information from branches is extracted, transformed, and loaded into a staging area of the EDW. After all the transactional data has been processed into a form, the transactional data can be loaded into the base tables of the EDW. The data in the staging area can be viewed as the delta of the base tables.
2. The delta of the base tables in the staging tables is propagated to the base tables. The indexes of the base tables are updated accordingly as well.
3. MQTs designed for the ad hoc queries (i.e. schema-driven) and their indexes are refreshed and updated for the base table delta
4. MQTs designed for the batch query workload (i.e. workload-driven) and their indexes are refreshed and updated for the base table delta. Note that the step 3 and step 4 can be interchanged.
5. Batch query workload is executed.

We illustrate the operational flow of an enterprise data warehouse in Figure 1. Note that in this figure, the arrow from the box of batch MQT refresh to the box of batch workload process indicates that the MQTs are used by the batch workload. Similarly, the arrow from the box of ad hoc MQT refresh to the box of ad hoc query processing indicates that the MQTs are used by the ad hoc queries, but the cost is added to the batch window. For enterprise data warehouse operations, there are two key requirements for the system performance. There are

- **Deadline to complete the batch workload:** The batch workload must be completed within a window. For example, the batch workload may start from the midnight and must be completed before 8am, when the business performance analysis reports need to be made available for the management. In order to run the batch workload, it is required to propagate the transactional data in the staging area to the base tables, and to refresh MQTs first. In addition to these tasks associated with the batch workload, MQTs designed for the ad hoc queries also need to be refreshed in the evening. The deadline to complete the batch workload could be defined as *six hours window* or *to start at 1am and to complete by 8AM*.
- **QoS requirement for the average response time for ad hoc queries**: Since it is not feasible to predict the patterns of the ad hoc queries that may be issued by users. Cube Views Optimization Advisor[4] is designed to recommend MQTs to improve the performance of the ad hoc queries. The MQTs recommended by Cube Views Advisor is based on schema and it recommends MQTs to *bridge* tables to improve join operations. With more disk space allocated, Cube Views Advisor

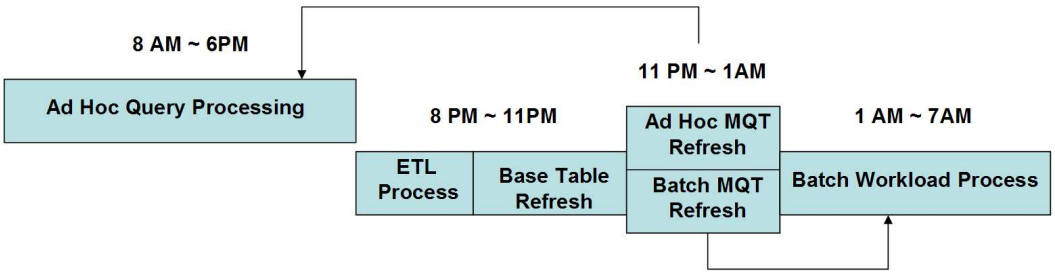

Figure 1: Operational Flow of Data Warehouse

can recommend more MQTs to cover a higher percentage of join operations for the ad hoc queries (probability wise). Users of the enterprise data warehouse like to have a QoS requirement defined as the average response time for the ad hoc queries.

2. CHALLENGES

To improve the performance of query processing in the data warehouse operation, many vendors have MQTAs, including IBM DB2 [1, 3], RedBrick/Informix [5], Oracle 10g [6], and SQL Server [7].

The MQTA in IBM DB2 is called the DB2 Design Advisor. It serves as the MQTA in this paper. This advisor was first created to provide index recommendations. It was later enhanced to recommend MQTs as well [2, 3]. The MQTs generated by the advisor includes multi-query optimization (MQO) techniques in which common expressions are found by compiling multiple queries at the same time [8]. The advisor also allows for a wide range of MQTs to be selected and maintained. In particular, we support full refresh MQTs (which are updated periodically by the user) and immediate refresh MQTs (which are updated whenever the base tables are updated), and impose no restrictions on the complexity of the queries that define these MQTs.

We now discuss the dependency among steps described in Section 1 and discuss the issues and complexity in designing an enterprise data warehouse to meet both deadline of the nightly batch workload completion window and the QoS requirement for ad hoc queries.

The dependencies between the operations listed in the previous section can be represented in Figure 2. The x-axis is the available disk space (or the number of MQTs). The y-axis is the elapsed time. Curves 1 to 4 are base table delta update time, batch MQT refresh time, ad hoc MQT refresh time, and batch query processing time. Curve 5 is the sum of curve 1 to 4, which is the total time needed to complete the batch operations. Line 6 indicates the upper bound of the batch window.

The more MQTs for batch queries deployed, the better performance the batch queries have. But the MQT refresh time goes up with increase in the number of MQTs. Similarly, the more indexes deployed for the batch queries, the better performance of the batch queries, but the cost for index maintenance is more due to the delta propagation. There is also a trade-off between ad hoc queries and batch operations. To meet the QoS requirement of ad hoc queries, we need to create MQTs for ad hoc queries. But these MQTs are refreshed in the batch widow thus, make it longer to complete the batch operations.

As shown in Figure 2, the total time to complete the batch operations will drop when the MQT and index maintenance cost is less than the benefit and increase when the MQT and index maintenance cost surpasses the benefit. The challenge is to recommend

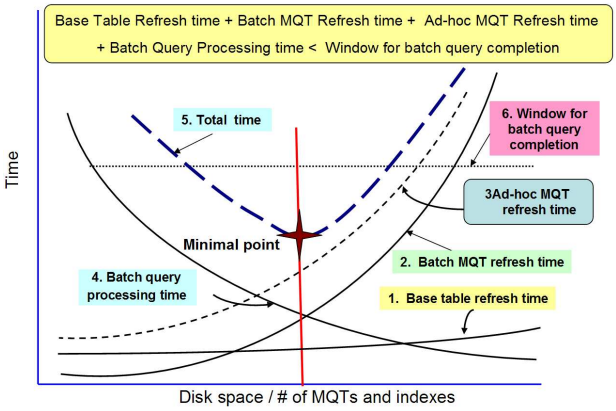

Figure 2: Dependency of Data Warehouse Operation Steps

a set of MQTs and indexes to complete the batch operation within the batch window while also meeting the QoS requirement for ad hoc queries.

This kind of solution may or may not exist for a given combination of the batch window and the QoS requirement for ad hoc queries. For example, if the QoS requirement is high which means the expected average response time for ad hoc queries is very short, more MQTs need to be created for ad hoc queries. The refresh time for these MQTs is added to the total time of the batch operations. If the refresh time is long enough, the optimal point of the total time (curve 6) is higher than the batch window upper bound. In this case, there is no solution meeting both requirements. Determining whether the solution exists is very helpful for the data warehouse users as this indicates better software and/or hardware configuration. To the best of our knowledge, there is no single tool solving the problem above.

3. SYSTEM ARCHITECTURE

To overcome these challenges above, we designed a *Data Warehouse Operation Advisor* on top of DB2 Design Advisor and Cube Views Advisor. The system architecture of the *Data Warehouse Operation Advisor* is shown in Figure 3. *Data Warehouse Operation Advisor* includes a capability planner, a cost simulator, and a calibrator. The workflow of *Data Warehouse Operation Advisor* is as follows.

- The capability planner receives the input parameters and sends the QoS requirement and the batch window upper bound (both are in elapsed time) to the calibrator.
- For the data warehouse operations that the capability planner has historic data (including query execution time with

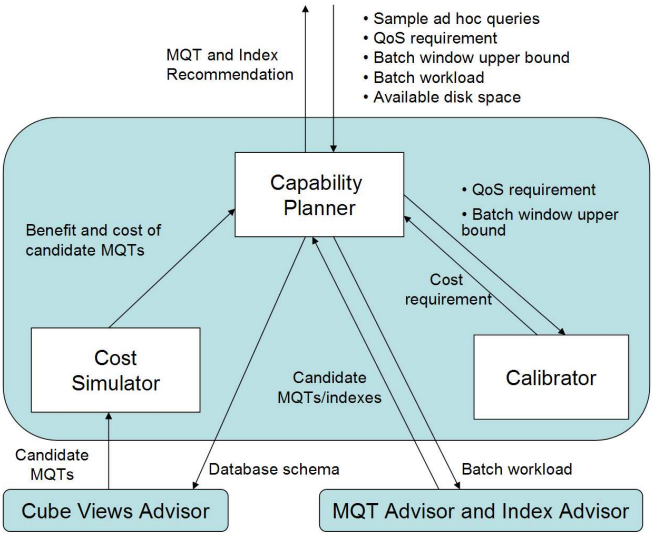

Figure 3: System Architecture of Data Warehouse Operation Advisor

and without MQTs, MQT refresh time, MQT size, base table refresh time, etc), the capability planner uses historic data based on real runs and applies time decay factors. For other operations that the calibrator has never seen, the capability planner communicates with the calibrator to calculate their expected execution times (in elapsed time) by multiplying cost requirement (in resource time unit) used by query optimizer by an operation specific calibration factor. More details are described in [9].

- The capability planner calls the Cube Views Advisor to generate MQT candidates which are then sent to the cost simulator for estimation of the cost and benefit. This cost information is returned to the capability planner. The capability planner recommends the smallest number of MQTs that meet the QoS requirement based on the ROI of the MQT candidates.
- The capability planner invokes the DB2 Design Advisor to generate the MQT and index candidates for the batch workload together with the cost information.
- Based on the cost information from Design Advisor and the recommendation of MQTs for ad hoc queries, the capability planner is able to determine whether a solution exists for the given requirements. If it exists, the capability planner calculates the best solution and outputs the recommendation.

4. DATA WAREHOUSE OPERATION ADVISOR

We implemented a prototype system of Data Warehouser operation Advisor. Figure 4 is a sample screen shot of the demonstration system. It takes a set of input parameters, conducts cost-based analysis and visualizes the capability planning results on the screen.

The input parameters shown in the upper area of the screen are described as follows.

1. Set the base table delta (percentage change in data). 1a shows the delta in current configuration and 1b is the delta in new configuration.
2. Set the load factor (percentage change in workload). 2a and 2b show the load factor in current configuration (fixes at 100%) and in new configuration respectively.
3. Set the batch operation completion time (in hours). See 3a and 3b.
4. Set the ad-hoc query average response time (in minutes). See 4a and 4b.
5. Choose which kind of queries to give priority. The chosen kind of queries will use up the remaining time window (when requirements are met), or use most of the time window anyways (when requirements are not met). A user can choose whether to give priority to ad-hoc query average response time or to batch workload completion time.

After setting the input parameters, a user can click the Analyze button to invoke the capability planning. The Operation Advisor will perform the cost-based analysis on the MQT and index candidates to determine whether the requirements can be met, and recommend the set of MQTs and indexes for deployment for optimal data warehouse performance. The following results are displayed in the lower area of the screen.

- Batch operation completion time. 7a shows the comparison of the current configuration with (i) the new configuration with provisioning, and (ii) the new configuration without provisioning. 7b, 7c, and 7d demonstrate the breakdown of batch operation completion Time for current configuration, new configuration without provisioning, and new configuration with provisioning.
- Ad hoc query average response time. The figure labeled 8 is the comparison of the current configuration with (i)the new configuration with provisioning, and (ii) the new configuration without provisioning
- Summary statistics. The area labeled 9 shows the summary statistics for queries, MQTs, indexes, and routing rates.

If the results show that the performance requirements can not be met, a user can change the input parameters and repeat the cost-based analysis until the results meet the requirements. Then the user can click on the OK button to move to the next step and deploy the MQTs and indexes.

In Data Warehouse Operation Advisor, we allow the user to do capability planning. The user (DBA or DW operator) can conduct what-if analysis. In the illustrated scenario, the batch window is 6 hours and the QoS requirement in terms of average response time for ad hoc queries is 15 minutes. In this window dump, we show the user who envisions the transaction volume will increase from 1% to 3% and system load will increase 40% during the last week of a quarter. The DW operator likes to plan ahead and make sure the DW can handle the increase of the workloads.

We can see without provisioning, the DW cannot complete its task within the required batch window. In this window dump, the advisor actually suggested to drop some indexes to improve delta loading time and drop some MQTs supporting ad hoc queries. We do expect the response time to increase for the ad hoc queries (advisor calculates how many MQTs should be dropped while still meeting QoS requirement).

5. CONCLUDING REMARKS

Data Warehouses and Business Intelligence (BI) applications are the top two priorities for CIO/CTOs. We have prototyped a Data Warehouse Operation Advisor to assist enterprise IT staff in configuring MQTs and indexes for both ad hoc queries and batch workloads. Our system is able to consider batch window deadline and QoS requirement for ad hoc query response time into consideration.

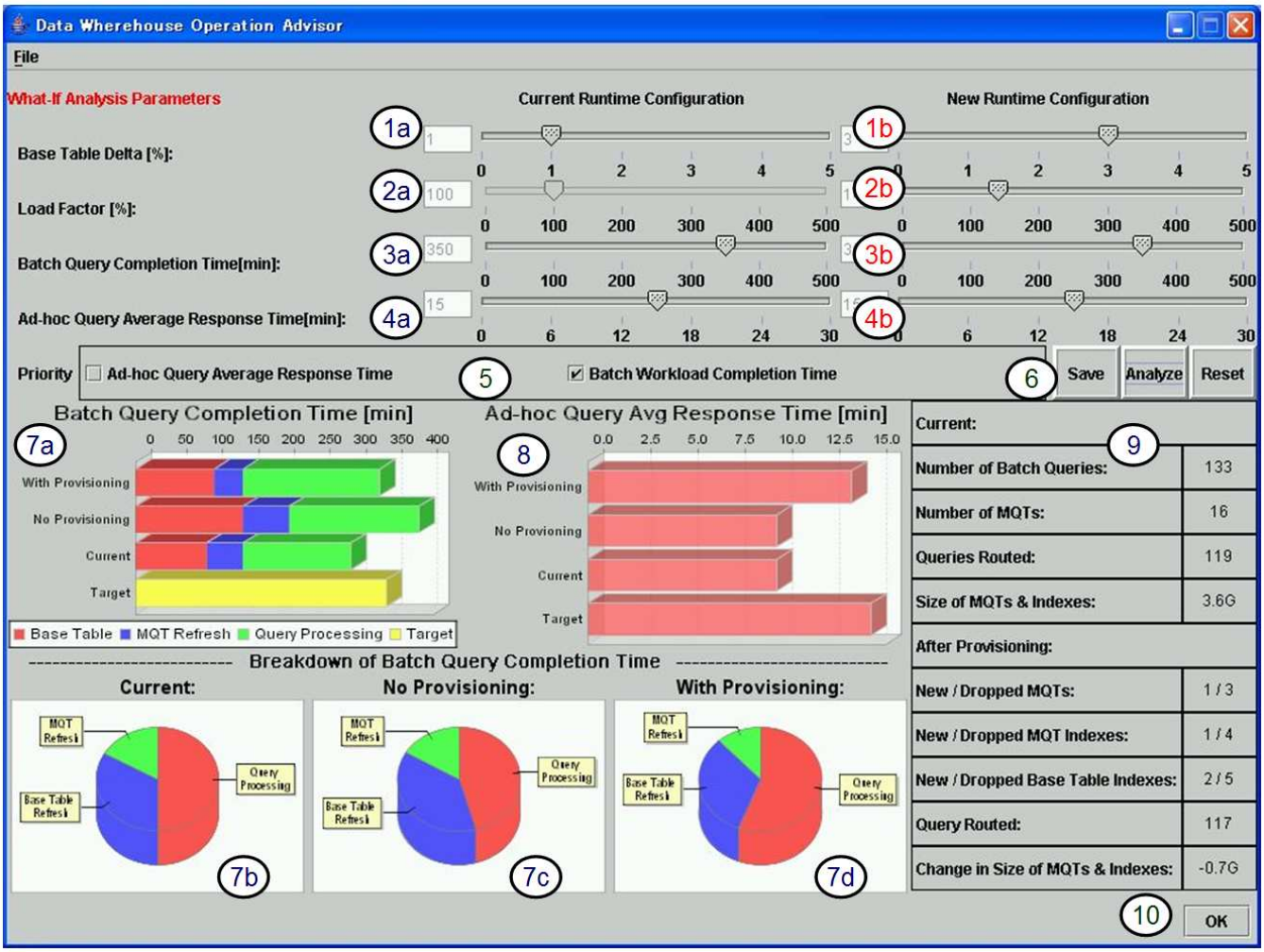

Figure 4: Window Dump of Data Warehouse Operator

Furthermore, our Data Warehouse Operation Advisor allows the data warehouse IT staff perform capability planning for foreseen or potential peak workloads. In this demonstration description, we highlight the key features of the Warehouse Operation Advisor. It shows a real and useful tool for data warehouse operations and high performance BI applications.

6. REFERENCES

[1] Gary Valentin, Michael Zuliani, Daniel C. Zilio, Guy M. Lohman, and Alan Skelley. DB2 Advisor: An Optimizer Smart Enough to Recommend Its Own Indexes. In *Proceedings of the International Conference on Data Engineering*, pages 101–110, 2000.

[2] Daniel C. Zilio, Jun Rao, Sam Lightstone, Guy M. Lohman, Adam Storm, Christian Garcia-Arellano, and Scott Fadden. DB2 Design Advisor: Integrated Automatic Physical Database Design. In *Proceedings of the International Conference on Very Large Data Bases*, pages 1087–1097, 2004.

[3] Daniel C. Zilio, Calisto Zuzarte, Sam Lightstone, Wenbin Ma, Roberta Cochrane Guy M. Lohman, Hamid Pirahesh, Latha S. Colby, Jarek Gryz, Eric Alton, Dongming Liang, and Gary Valentin. Recommending Materialized Views and Indexes with IBM DB2 Design Advisor. In *Proceedings of the International Conference on Autonomic Computing*, pages 180–188, 2004.

[4] IBM Cube Views Optimization Advisor. *Information available at* `http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0511pay/`.

[5] Red Brick. http://www.informix.com/informix/solutions/dw/redbrick/vista/.

[6] Oracle Corp. http://www.oracle.com/.

[7] Sanjay Agrawal and Surajit Chaudhuri and Vivek R. Narasayya. Automated Selection of Materialized Views and Indexes for SQL Database. In *Proceedings of the International Conference on Very Large Data Bases*, pages 496–i–505, 2000.

[8] Wolfgang Lehner, Roberta Cochrane, Hamid Pirahesh, and Markos Zaharioudakis. Fast Refresh using Mass Query Optimization. In *Proceedings of the International Conference on Data Engineering*, pages 391–398, 2001.

[9] Wen-Syan Li, Vishal S. Batra, Vijayshankar Raman, Wei Han, Kasim Selcuk Candan, and Inderpal Narang. Load and Network Aware Query Routing for Information Integration. In *Proceedings of the International Conference on Data Engineering*, pages 927–938, 2005.