# Technique for selection of materialized views in Data warehouse based on costs model

B.ASHADEVI
Assistant Professor / MCA, Velalar college of Engg and Technology, Thindal, Erode – 638 009
Tamilnadu, Tel-No : 09486363888
asharajish2005@gmail.com

Dr.R.BALASUBRAMANIAN
Associate and Dean of computer Applications, Sri Krishna College of Engg and Technology, Kuniamuthur, Coimbatore -8, Tamilnadu.

## ABSTRACT

A data warehouse stores materialized views of data from one or more sources for the purpose of efficiently implementing decision support or OLAP queries. One of the most important decisions in designing a data warehouse is the selection of materialized views to be maintained at the warehouse. The goal is to select an optimum set of materialized views that minimizes the sum cost of processing set of queries. In this paper, a new Technique for the selection of optimum set of materialized views is proposed. The proposed technique establishes this set based on the utilization rate, calculation cost, and the relationship update cost regarding the operator type (Select, Project or join) and it also includes the query evaluation cost.

In this proposed technique, for the utilization rate, the optimum profit, obtained with the Greedy algorithm, is multiplied by total number of dependent relationships, because the utilization rate has a straight rate with the number of dependent relationships. The view calculation cost has been considered to decrease in proportion to the number of dependent relationships, this caused by the fact that the calculation cost can be shared between them, so the calculating cost is divided by total number of dependent relationships. In the base relationships update rate, the number of elements inside the view that can change or evolve has to be established. The proposed algorithm also takes into account the query evaluation cost. In this case, the order of operator type (Select, Project or join) has to be established.

## Keywords

Materialized views, greedy algorithm, utilization rate, calculation rate, relationship update cost, query evaluation cost.

## 1 INTRODUCTION

A data warehouse is a repository of integrated information available for querying and analysis. The information stored at the warehouse is in the form of views, referred to as materialized views, derived from the data in the sources. The design and construction of a data warehouse are complex and thoughtful tasks, made of several processes commonly called: extraction-integration, organization and querying. For extraction, the inner and outer data sources must be analyzed. This analysis is useful to both the selection of the data to be store in the warehouse, and the selection of the tools needed for the extraction and transformation of these data before they are stored. The second process consists of the organization of these data inside the warehouse. To accomplish this, the multidimensional model must be designed, and the optimum set of materialized views must be defined. Finally, the last process, the querying, establishes the required tools for the displaying of the data set [2].

The paper is focused at the data warehouse structure based on the internal organization process. In [2], establishes the optimum set based on the utilization rate, calculation cost, and the relationship update cost regarding the operator type (Select, Project or join). The constellation schema, hypercube materialization and two algorithms were described for the selection of the optimum set of materialized view.

In this paper, the optimum set of materialized view is based on utilization rate, calculation cost, and the relationship update cost regarding the operator type (Select, Project or join) and also the query evaluation cost. The rest of this paper is organized as follows. Section 2 explains a new technique for the selection of optimum set of materialized view based on query evaluation cost. Section 4 presents conclusions and future work.

## 2 MATERIALIZED VIEW SELECTION

In [2], Hypercube materialization was developed to select the materialized view, the problem is reduced to establish the dependent cells set to materialize. The Greedy algorithm proposed in [1][3] is based upon a cost model to determine the optimum set of materialized views. It uses the storage costs and the number of dependent views (optimum) of a view to calculate its optimum profit.

In [2], the developed algorithm described the dependent relationships play a key role. Here, the number of dependent relationships of a view is an initial parameter for utilization rate and the calculation cost. For the utilization rate, the optimum profit, obtained with the Greedy algorithm, was multiplied by total number of dependent relationships, because the utilization rate has a straight rate with the number of dependent relationships. The view calculation cost was considered to decrease in proportion to the number of dependent relationships, this caused by the fact that the calculation cost shared between them, so the calculating cost is divided by total number of dependent relationships. The developed algorithm also takes into account the base relationships update rate. In this case, the number of elements inside the view that can change or evolve was established.

An extended technique of greedy algorithm using the cost analysis methodology for evaluation is then presented for selecting an optimal set of materialized views.

## 2.1 COST MODEL

In [5], the estimated query, maintenance and storage costs in the following descriptions were calculated in terms of data block size B.

### *2.1.1 QUERY PROCESSING COST FOR SELECTION, AGGREGATION AND JOINING*

The analysis assumes that there is no index or hash key in any of the summary views, therefore linear search and nested loop approach are used for the selection and join operations, respectively. The total query cost *Total(Cqr)* for processing *r* user's queries between each update time interval is:

$$Total\ (C_{qr}) = \sum_{i=1}^{r} f_{qi} * C_q(q_i)$$

### *2.1.2 DATA WAREHOUSE MAINTENANCE COST*

Assume that re-computation of each summary view $V_i$ requires selection, aggregation and joining of its ancestor view $V_{ai}$ with n dimension tables. If there are j summary views in the warehouse which are materialized, the total maintenance cost 'Total($C_m$)' for these materialized views is then :

$$Total(C_m) = \sum_{i=1}^{r} f_{ui} * C_m(V_i)$$

*( $f_{ui}$ = 1 in [5], since we assume that all sales summary views are updated once within a fixed time interval.)*

### *2.1.3 STORAGE COST*

Storage cost of summary view $V_i$ in terms of data block B is

$$C_{\text{store}}(V_i) = S(V_i)$$

### *2.1.4 THE NET BENEFIT AND COST EFFECTIVENESS*

In order to determine the set of optimal materialized summary views, the net benefit 'Net($B_i$)' and the storage effectiveness '$n_i$'(i.e the net benefit per unit of storage space occupied by a materialized view) associated with each summary view have to be calculated, as follows:

Storage effectiveness of each summary view $V_i$ is calculated as follows:

$$\text{Net}(B_i) = \{\sum_{i=1}^{r} f_q\ (V_{ni}) * [C_t(V_{ni} \leftarrow V_{ai}) - C_t(V_{ni} \leftarrow V_i)]\} - C_m(V_i) - C_{\text{store}}(V_i)$$

$$n_i = \text{Net}(B_i) / S(V_i)$$

The storage effectiveness $n_i$, net benefit Net(Bi), storage cost $C_{\text{store}}(V_i)$, maintenance cost $C_m(V_i)$, query frequencies $f_{qi}$ and the total cost $C_{\text{total}}$ of summary views $V_i$ are calculated and listed in [5]

## 2.2 EXTENDED TECHNIQUE OF GREEDY ALGORITHM FOR MATERIALIZED SUMMARY VIEW SELECTION

Let *T* be the set of all sales summary views grouped by various dimension key attributes. Based on the greedy algorithm of [1], we develop an adapted greedy algorithm for determining the optimal set of materialized summary views *L*, a subset of *T*, such that the total cost $C_{total}$ is minimized. The algorithm is based on the cost model presented in section 2.1.

### *2.2.1 MATERIALIZED VIEWS SELECTION TECHNIQUE:*

1. Determine the optimum query and maintenance paths for computing all summary views in the data warehouse;
2. Calculate the Net(Bi) and ni of each summary view in the query paths. Let T be the number of summary views possibly chosen as materialized views.

***for*** *i =1 to T* ***do*** Calculate the *Net(Bi)* of each summary view

$$\text{Net}(B_i) = \{\sum_{i=1}^{r} f_q\ (V_{ni}) * [C_t(V_{ni} \leftarrow V_{ai}) - C_t(V_{ni} \leftarrow V_i)]\} - C_m(V_i) - C_{\text{store}}(V_i)$$

Storage effectiveness of summary views:

$$n_i = \text{Net}(B_i) / S(V_i)$$

3. List summary views in descending order according to the value of their storage effectiveness such that those views with the best storage effectiveness will be chosen first;
4. Calculate the $C_{total}$ for each view :

```
        $i = 1$;
        $C_{total} = Total(C_{qall}) - Net(Bi)$;
        for $i = 2$ to $T$ do $C_{total} = C_{total} - Net(Bi)$;
        find the $Min(C_{total})$ as the optimal cost for materialized
        view selection;
5.  Select the best materialized view set L
        $i = 1$;
        $C_{total} = Total(C_{qall}) - Net(B_i)$;
        while $C_{total} > Min(C_{total})$
        $i = i + 1$;
        while $S(L) < S$
        Select $V_i$ from the summary view set $TL$
        with the highest storage
        effectiveness;
        $S(L) = S(L) + S(V_i)$;
        endwhile
        $C_{total} = C_{total} - Net(B_i)$;
        endwhile
        return L.
```

where *L is t*he set of optimal materialized view in [5]

# 3 CONCLUSION AND FUTURE WORK

An Extended technique of greedy algorithm using the cost analysis methodology for evaluation was developed for selecting an optimal set of materialized views. The total cost evaluated under the *partial-materialized-views* method. The partial-materialized-views method requires the shortest total processing time. The proposed technique has a drawback, there was no index or hash key in any of the summary views, therefore linear search and nested loop approach were used for the selection and join operations, respectively. It discards the query evaluation cost regarding the operator type such as duplicate elimination, aggregation, joins, and outer joins. Hashing and sorting are dual, in the sense that many operations such as duplicate elimination, aggregation, joins, and outer joins can be implemented via either hashing or sorting.

# 4 REFERENCES

[1] H.Gupta. "Selection of views to materialize in a Datawarehouse". Proceedings of 23rd VLDB conference, Athens, Greece 1997.

[2] M.T.Serna-Encinas, J.A.Hoyo-Montano,"Algorithm for selection of materialized views based on a costs model.

[3] H.Gupta and I.S.Mumick, "Selection of views to materialize under a maintenance cost constraint"

[4] V.Hariharan, A.Rajaraman, and J.Ullman. "Implementing data cubes efficiently", Proceedings of ACM SIGMOD 1996.

[5] G.Chan, Qing Li, Ling Feng. Design and selection of materialized views in a data warehousing environment: A case study .1996