Fractionalized View Materialization in Data Cubes¹

Yon Dohn Chung, Woong Je Park, Min Jung Kim, Myoung Ho Kim

Division of Computer Science, Dept. of EECS.

Korea Advanced Institute of Science and Technology

373-1 Kusong-dong, Yusong-gu, Taejon 305-701, Korea

E-Mail: {ydchung, ujpark, mjkim, mhkim}@dbserver.kaist.ac.kr

Abstract

In this paper, we introduce the concept of the 'fractionalized view'. The fractionalized view is the result of a query defined through aggregate functions and the range on each dimension, where the subdivision of each dimension is based on queries' access patterns. For the representation and selection of fractionalized views to materialize, we define the 'AND-OR cube graph'. With this structure, we propose a fractionalized view materialization method

1. Introduction and Motivation

The view selection is to pre-compute some views in the data cube and store them to efficiently process queries in the future, for the access to pre-computed views is more efficient than the access to the base database. Therefore, the view selection is very important to the overall system performance.

In the previous view selection approaches, they consider a node in the lattice as the selection unit. However, if the query access patterns are not uniform in a node, the previous approaches are not effective. In practice, the query access patterns concentrate on some hot regions of attributes, thus the node in the lattice framework has to be organized to incorporate the preference regions of the queries. Figure 1 shows an example that a user query accesses the data cube in a non-uniform pattern. The attribute P is the title of a phonograph record, the attribute C is the name of a customer who buys a record. In the example, we see that the records are classified into categories and the customers are classified into age groups. Also, the teenager customers prefer the pop music. Thus, in this application, we can improve the system performance by materializing the parts for teenager customers and pop records, not the entire node that represents all records or all customers.

Figure 1: An Example of Non-uniform Access Pattern

2. The Proposed Method

In the paper we assume that attributes are organized in hierarchies and the region of each attribute is divided into sub-regions according to the queries' access patterns [6]. We call the data cube, where each attribute is divided, the *fractionalized data cube*, and the view that is generated from the combination of divided attribute regions the *fractionalized view*. And, we propose a fractionalized view materialization method. In the method we define the *AND-OR graph* for representing the dependence relation between fractionalized.

Figure 2 shows an example of fractionalized data cube where there are two attributes D (date) and P (product), and both attributes are divided into two sub-regions and are organized in single-level hierarchies. Thus, there can be 9 fractionalized views (denoted by n0, n1, ..., n7 and n8 in the figure) in the fractionalized data cube. Figure 3 shows the *fractionalized cube* of Figure 2 and its *AND-OR cube graph* representation. The basic concept and notation are similar to those in [5].

We use a greedy method for selecting nodes to materialize in the *AND-OR cube graph*[6]. In each step, we select a node that minimizes the total query cost considering its required storage space.

¹ This work was supported by grant No. 1999-1-303-007-3 from the interdisciplinary Research program of the KOSEF

Figure 2: The Division of Attributes' Regions and Fractionalized Data Cube

Figure 3: AND-OR Cube Graph

3. Performance Evaluation

In the experiments, we use a data set with three dimensions, each dimension consists of 3-level hierarchy and is divided into two sub-regions. We assume the data objects are uniformly distributed in the data cube and we use the estimation methods in [3] for configuration of the number of data objects. The configuration of query patterns is as follows:

- Query access patterns for hot regions (QX): The X % of the queries access the hot 20% regions, and the other queries access the data cube in a uniform fashion.
- Actual query access patterns (PX): The actual query access patterns coincide with the estimated ones in X %.

4. Conclusion

In this paper, we have proposed a *fractionalized view materialization* method. The *fractionalized view* is determined by the combination of aggregation attributes' sub-regions, not entire regions of the attributes. The division of an attribute into sub-regions is based on the queries' access patterns.

Figure 4: Performance Improvement According to Hot Region Accesses

Figure 5: Performance Improvement According to Actual Query Sets

For the fractionalized cube representation, we have defined the *AND-OR cube graph* structure. With experiments, we have evaluated the performance of the proposed method compared with the conventional method. In the results, we have found the proposed gives better performance than the conventional one.

References

- [1] V. Harinarayan, A. Rajaraman, and J. C. Ullman. Implementing data cubes efficiently. In Proceedings of ACM SIGMOD, pages 205-227, 1996.
- [2] A. Y. Levy, A. Rajaraman, A. O. Mendelzon, Y. Sagiv, D. Srivastava. Answering queries using views. In Proceedings of ACM PODS, 1995.
- [3] K. A. Ross, D. Srivastava. Fast Computation of Sparse Datacubes. In Proceedings of the 23rd International VLDB Conference, pages 116-125, 1997.
- [4] A. Shukla, P. M. Deshpande and J. F. Naughton. Materialized View Selection for Multidimensional Datasets. In Proceeding of the 22nd International VLDB Conference, pages 522-531, 1996.
- [5] H. Gupta, Selection of views to materialize in a data warehouse, In proceedings of ICDT, pp.98-112, 1997.
- [6] M. J. Kim. Fine Granule View Materialization in Dimension Hierarchy-based Data Cube, MS Thesis, KAIST, 2000.