International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 14 (2014), pp. 1411-1417

http://www. irphouse.com

New Approach of Computing Data Cubes in Data Warehousing

Prabhjot Kaur[1] and Puneet Kaur[2]

[1] Guru Nanak Institute of Management & Information Technology, New Delhi
[2] Student, IGNOU, New Delhi

Abstract

The paper is dealing with data cubes built for data warehouse for OLAP purposes. OLAP (Online Analytical Processing) system offers multidimensional data analysis in which large volume of historically collected data is computed. To decrease the query time and to provide various options to the analysts, a data model was designed to organize data perfectly in a multidimensional data model. In OLAP, a complex query may result in many scans of the base table, leading to poor performance. Because sub totals are very common in OLAP queries, so it is desired to define a new operator for the collections of such sub totals namely data cube structure. Data cube is a cube shaped multidimensional model in which each cell of the cube corresponds to a set of values for different dimensions. It consists of core or base cuboids surrounded by a collection of sub cubes or cuboid that represents the aggregation of base cuboids along one or more dimensions. In total a d-dimensional data warehouse is associated with 3/4 d sub cuboids. In practice, they are normally pre-computed so as to improve the efficiency of user query. We describe cluster based implementation of an algorithm to compute this multidimensional model named data cube. Though a number of efficient sequential algorithms have recently been proposed for this problem, very little research effort has been expended upon cost- effective parallelization techniques. Our approach builds directly upon existing sequential proposals and is designed to cover load balance and makes the communication effective and efficient.

Keywords: Data cubes; Data warehouse; OLAP; Multidimensional database; Parallelization techniques; Sequential algorithm; Query processing

Introduction

A data warehouse is a relational database that is designed for query and analysis which serves the purpose of decision support. Over the last year, we have seen tremendous growth in data warehousing market. Despite the sophistication and maturity of conventional database technologies, the ever increasing size of corporate databases, coupled with the emergence of the new global internet "database", suggests that new computing models may soon be required to fully support many crucial data management tasks.

In our current research, we focus on "data cube" which is a database operator that is used to pre compute multiple views of selected data by grouping the values in possible combinations using 'group by' clause in SQL. The resulting data will then be used to response the query of the user. In sequential computation, a significant work is carried out on computing the data cube which has resulted in many algorithms that are used to reduce the computation time. By contrast, relatively very little research effort is done on parallel computation of various possible views of data cubes. While parallel computation provides better performance than sequential computation but it requires excessive inter node communication.

Our approach is to divide the data cube views into individual nodes first and then they are independently computed using sequential algorithms. In our current research we had tried to explain the concept data cube with the help of cross tabular illustrations. We start from the concept of OLAP (Online Analytical Processing) and considering its limitations. It is desired to define a new operator for collection of subtotals namely Data Cubes.

Background

Data warehouses can be described as Decision Support Systems in which they allow users to access the evolution of an organization in terms of number of key attributes or dimensions. These attributes are extracted from various operational sources and then cleaned and normalized before being loaded into a relational store. By exploiting multi dimensional views of the underlying data warehouse, uses can "drill down" or "roll up" on hierarchies, "slice and dice" particular attributes or perform various statistical operations such as ranking and forecasting. This approach is known as OLAP (Online Analytical Processing). Figure 1 illustrates the basic model.

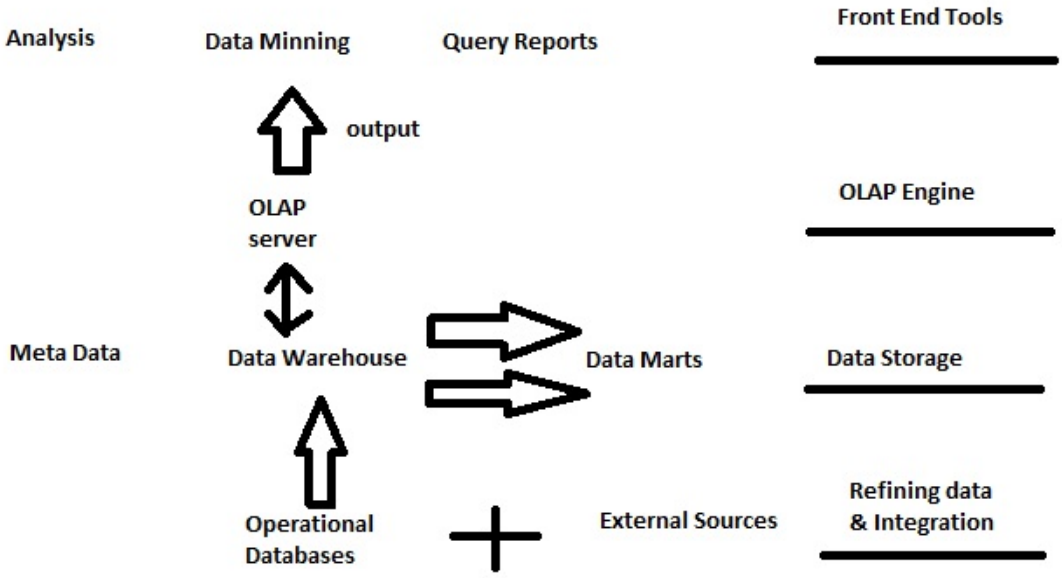

Figure 1: Basic OLAP diagram

OLAP is mainly used for analyzing business data collected from daily transactions. The main purpose of the OLAP system is to enable analyst to construct mental image about perspective data by exploring it from different perspectives at different levels of organization in interactive manner.

The data is usually organized in the form of relational model known as Star Schema which stores multi dimensional data. Figure 2 shows the basic building of star schema. Each dimensions table contains its specific information and fact table which contains fact full information that is information which is of analysts interest and which correlates all the dimension tables. Figure 3 illustrates one example of star schema which has fact table (time_id, college_id, salary) where first two attributes time_id, college_id are called dimensions and salary is called measure. Each dimension has dimension table associated with it. The dimension table may contain redundancy, which can be removed by splitting each dimension table into multiple tables, one per attribute in dimension table. The result is called Snowflake Schema.

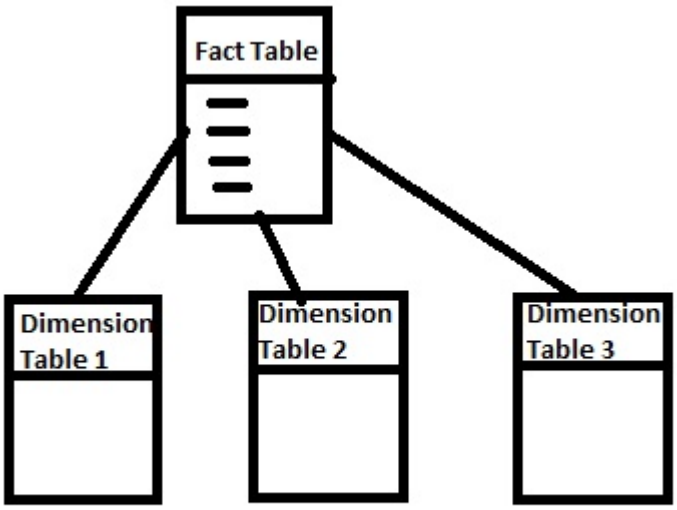

Figure 2 : Star Schema

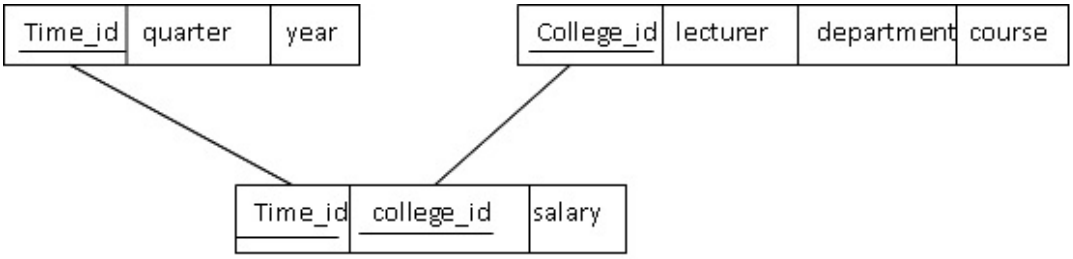

Figure 3: An example of Star Schema

Data cube

It is convenient to consider the data as an n-dimensional cube because the views in a data warehouse are of multidimensional nature. Basically a data cube consists of a core cuboids which is surrounded by a collection of sub cubes or cuboids which is aggregate as one or more dimensions. As a result, d-dimensional data warehouse is associated with ¾ d cuboids.

Building the data cube

A fully materialized data cube consist of a ¾ d individuals views. We normally use our conventional SQL to construct tables of multidimensional data cube. Data cube was proposed as a SQL operator to support common OLAP tasks like histogram and subtotals. Even though such tasks are usually possible with standard SQL queries, the queries may become very complex.

Table 1 : Base Relation 'comp'

LECTURER	QUARTER	DEPARTMENT	SALARY
Franklin	Q1	IT	20, 000
Franklin	Q1	Management	25, 000
Bob	Q1	IT	15, 000
Charlie	Q1	IT	20, 000
John	Q1	Management	30, 000
Franklin	Q2	IT	25, 000
Bob	Q2	IT	150, 000
Charlie	Q2	It	10, 000
John	Q2	Management	20, 000
John	Q2	Management	15, 000

For example Table 1 shows a simple relation ‘comp’ where lecturer, quarter and department are three dimensions and salary is a measure. We call ‘comp’ a base relation since it contains the ground fact to be analyzed. Subtotals salary in the 2-D cross table is shown in Table 2. The inner tabular includes the quarterly computed salary of each lecturer, below the inner tabular is the total salary for each lecturer, to the right hand side is the total salary in each year, at the right bottom corner is the total salary of all lecturers in the two years.

		Franklin	Bob	Charlie	John		Total
Q1		45, 000	15, 000	20, 000	30, 000		1, 10, 000
Q2		25, 000	15, 000	10, 000	35, 000		85, 000
Total		70, 000	30, 000	30, 000	65, 000		1, 95, 000

Figure 2 : A 2-D Cross Table

Our approach to parallelize Data cube

In our current research, we have sought to develop load balanced and communication efficient parallel algorithms that in turn exploit the efficient of the existing sequential approaches. In our method of building data cubes, we partition the original problem into set of sub cubes whose computations are then distributed to individual processors. Our algorithms require very little communication overhead and are applicable to high dimension spaces.

Implementation

We have proposed parallel algorithms for both top-down and bottom up paradigms. There is a top down technique that attempts to find a set of short paths within the lattice such that the cost of computing child views from their parents is minimized. A subset of the child views from their parents is minimized. A subset of the child views

are computed using linear scan of parent view, while the remaining views require a resorting of a parent cuboids.

In parallel our task is to find partitioning of the lattice that balances the cost of sub cubes which are computed across p processors. We employ a partitioning algorithm which divides the network in a specific way. Although this procedure of partitioning does not guarantee an optimal split across the hardware, but it guarantee a lower bound and the size of largest subset. Once these subsets have been established, they are then distributed to the local nodes where existing sequential algorithm are executed.

Basic Algorithm

The algorithm consists of following steps:-

1. Construct a lattice with ¾ d views.
2. The size of each view is estimated.
3. Cost of using view is determined by computing its children, use the estimated size to calculate
    a. Cost of scanning the view
    b. Cost of sorting
4. Lattice is reduced into spanning tree using bipartite matching technique which identifies appropriate set of prefix ordered sort paths.
5. Spanning tree is then partitioned / divided into p sub trees.
6. Each sub tree is distributed to p compute nodes.
7. Each node is then build to set of views using sequential algorithm.

Future Work

There are many data cube related problems which are still remained to be addressed. In practice, data warehouse designers may wish to generate some subset of all available views. Given the potential advantages of bottom up approaches to data cube construction, it will be interesting to compare our parallel sorting algorithm with bottom up alternative. There is even a need to address the problem of querying the data in the parallel environment. Also the indexing mechanisms are required for both linear scanning as well as parallel scanning of cuboids for OLAP processing. Future work is intended in implementation of hierarchies for cube dimensions and the storage structure is required for supporting hierarchical dimensions. Processing of analytical queries need to be employed for selection of aggregates satisfying certain condition and optimized multilevel list search.

Conclusion

As the data warehouse is growing in size and complexity, so there are opportunities for researchers to provide powerful and cost effective OLAP solutions. In this paper we have discussed the implementation of parallel algorithm for constructing multi dimensional data storage model known as data cube. Our paper demonstrates the technique which is viable even in cluster environment. This storage structure

overcomes the problem with sparseness by storing existing data values only. Analytical queries and a formal description of OLAP query has been presented in this paper.

References

[1] Rozeva, A. *Index Structure for the Fact Table of a Star-Join Schema and Template Query Processing*., 2003
[2] Lima, Mattoso, A. A. B., *OLAP Query Processing in Data Cluster*, 2005
[3] Golferelli M., Maniezzo v., Rizzi S. *Materialization of Fragmented views in multidimensional databases*, 2004
[4] Furtado P., *Hierarichal aggregation in networked data management*, 2004
[5] Sarawagi S, Agarwal R, Gupta A, *Computing Data cube*, 1996
[6] Chaudhari, S., U. Dayal, *An overview of Data Warehousing and OLAP Technology*, 1997
[7] Information on http://searchdatamanagement. techtarget. com/feature/Data-warehouse-archietectures-concepts-and-phases