Distributed Data Cube Materialization mechanism

RAJIV KUMAR, Dr S.Venkatanarayanan
[1]Research Scholar
SSSUTMS, Sehore
[2]Research Guide
SSSUTMS, Sehore

ABSTRACT

Data Cube provides basic abstraction for multi-dimensional data analysis and allows you to discover useful information from a range of data. Online Analytical Processing (OLAP) has taken it to the next level by supporting online responses to analytical queries with the underlying technique that pre-calculates (realizes) the data cube. The data cube materialization is important to OLAP, but a costly task in terms of data processing and storage. It is important that a distributed data cube materialization mechanism be suitable for multidimensional data analysis across the current distributed storage systems. Several research papers are currently available that have considered MapReduce for the data cube materialization. In addition, Apache Spark recently enabled the CUBE operator as part of its DataFrame application.

Keywords: Data cube, Multidimensional array, MapReduce Merge, OLAP

INTRODUCTION

The Multidimensional Data Modeling does not originate from database technologies. As a manual data analysis tool based on multidimensional matrix algebra, it dates back to late 19th century. The basic notion of the multidimensional modeling is to define a data relationship model among all possible data attributes in a data collection. The possible data relationships appear in the dataset are Number of Sales per Model, Number of Sales per Year, Number of Sales per Model and Year, etc. Multidimensional data modeling classifies data attributes into two main categories; Dimensions and Measures. The measures represent quantitative attributes countable with the combination of dimensional attributes. In the sample dataset, Number of Sales represents a measure field, whereas Year, Model and Color stand for dimensional fields.Multidimensional data analysis is one of widely used data analysis approach in many enterprise application domains. The hypercube or more commonly known Data Cube provides a Multidimensional Analytical Model for data analysis. The data cube organizes factual information as Dimensions and Measures. The Measures represent numerical properties of factual information, which are explained and selected via associated dimensions. The high-level Cube Operations; Roll-up, Drill-down, Drill-across, Slice, and Dice allow querying multidimensional data for various data analysis questions. The On-Line Analytical Processing (OLAP) led data analysis into the next level providing fast query responses for complex data analysis questions.

MULTIDIMENSIONAL DATA ANALYSIS

Data Cube provides an adequate model for Multidimensional Data Analysis, which involves decision support queries that require heavy use of data aggregations. The OLAP brings Multidimensional Data Analysis to the next level supporting

timely response on decision support queries. Providing a swift response on OLAP queries require precomputation or materialization of all possible data cubes. But, Data Cubes are intrinsically large and grow exponentially with the number of dimensions and its cardinality level. Also, computationally expensive against the number of dimensions. These factors bring a new research problem to the community; Data Cube Materialization with efficient storage and computation.

Most of the early research proposals considered full cube materialization with efficient grouping algorithms [2]. With growing dataset and high dimensional data domains, the full cube materialization was determined inefficient and impractical. Therefore, several other approaches were proposed during the time. In one research direction, data cube compression was considered with models such as QuantiCubes [10], Quotient Cube, Condensed Cube, Closed Cube, Dwarf Cube and MDAG Cube. Another research direction was to select part of the cube for the materialization; Iceberg Cube, BPUS and PBS. With a naive assumption that data analysis does not require an accurate result, approximate data cube computation was engaged in types of research; Quasi Cube [11], Wavelets [13] and Loglinear Cube [14]. Also, some researchers improved previously proposed algorithms with parallel computation models. In general, early researchers on data cubes can be categorized into 4 main topics; Full data cube computation with efficient algorithms, Data cube compression, Partial cubes and Approximate cubes. The Data Cube operator and Lattice framework [14] provided the base semantic in all of these research directions.

FULL CUBE MATERIALIZATION

Early research work on data cube materialization considered the calculation of full cube. That is the materialization of all possible data cubes with efficient data grouping methods. It is the CUBE relational operator first initiate the idea of computing of all possible data cubes.

Multidimensional Arrays

In contrast to ROLAP approaches based on CUBE operator, Y. Zhao, P. M. Deshpande, and J. F. Naughton proposed data cube computation approach for Multidimensional OLAP systems, which store data in physical sparse arrays. The proposed MultiWay array methods resulted in better performances than previous ROLAP methods in term of both CPU and storage.The multidimensional array based techniques (MOLAP systems) found efficient in computing cuboids with the fast random accessing capability in arrays. But, it was a limitation that the size of the arrays at each dimension required to be fixed and need the reallocation of all elements in the arrays in case introduction of new dimensional value. T. Tsuji, A. Hara, and K. Higuchi proposed an incremental cube computation for MOLAP by employing extendible arrays, which can be extended dynamically in any direction without any data reallocation. D. Jin, T. Tsuji, and K. Higuchi took it to the next level using a single extendible array to manage full cube incrementally.

PARTIAL CUBE MATERIALIZATION

Many researches considered full data cube materialization does not scale in term of computation and storage space. The number of cuboids growth exponentially against the number of dimensions in a model. For example, consider a 3 dimensional model with dimensions d1, d2, and d3. The possible combination of group-by or cuboids are d1d2d3, d1d2,

d2d3, d1d3, d1, d2, d3 and all. In this way, the 3-dimensional model produces 8 possible dimensional combinations. That is n-dimensional model results computation of 2n number of cuboids. But in practice dimensions are expressed in a hierarchy of multiple levels. This even makes the calculation of more cuboids.Also, it is expensive in term of storage. The number of entries or tuples created from each cuboid depends on the cardinality of the cuboid.

$$\text{Total number of cuboids} = \sum_{i=1}^{i}(Li + 1)$$

This led to another research direction that consider the materialization of the subset of cuboids that is relevant for OLAP queries. The lattice framework made the foundation for this research direction.

The Lattice of Cuboids

The calculation of cuboids is an expensive process, and it is impractical to materialize all the possible cuboids. This led research direction to the partial materialization of the cubes. V. Harinarayan, A. Rajaraman, and J.D. Ullman proposed the lattice framework, which states that some cuboid can be computed from the result of another cuboid. They generalized the problem in a lattice diagram which shows dependencies of cuboids. The cuboid d1, d2, d3 is called based cuboid as the all other cuboids can be calculated from it.The lattice framework was attractive among many other researches, and it became the base approach for materializing cuboids. The authors of the paper [14] also contributed with a greedy algorithm, which selects best cuboids for materialization from the lattice framework.

BPUS, PBS Algorithms

The lattice framework [34] expresses dependencies between cuboids. V. Harinarayan, A. Rajaraman and J. D. Ullman with lattice framework, proposed a greedy algorithm, BPUS (benefit per unit space) that attempts to maximize the benefit of the set of cuboids selected. Later, A.Shukla, P. M. Deshpande, and J. F. Naughton proposed PBS (Pick By Size) algorithm, which runs several orders of magnitude faster than BPUS. It uses a chunk based precomputation, which has a benefit based cost model for chunks.

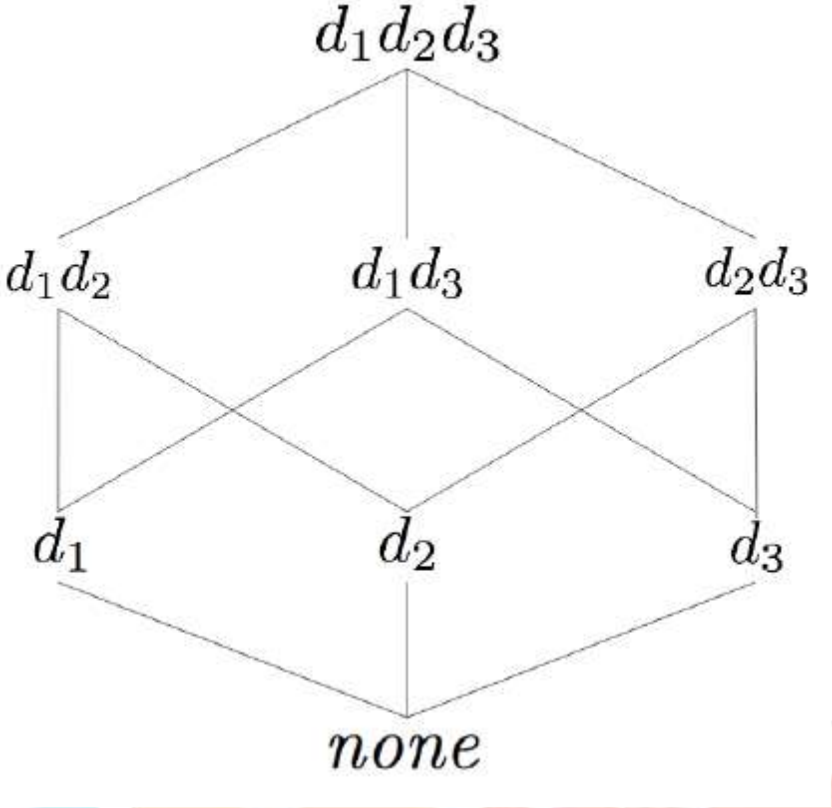

Figure 1.1: Lattice diagram of cuboid

Data Cube compression

Another research approach in data cube materialization is to find a more compressed model for the data cube to reduce large access overhead. Most of the compressed technique consider archiving the model, which cause data nonqueriable. The research problem is to find techniques that compressed the data model while keeping the full queriability and random access to the measures.

QuantiCubes

P. Furtado and H. Madeira proposed a data cube compression strategy called QuantiCubes, which uses a fixed-width compression coding to compress the data cube values.

Condensed Cube

WeiWang et al. proposed Condensed Cube that reduce size of the data cubesand hence achieve better performance. Even-though condensed cube reduces thesize, it is a fully computed cube and does not require further decompression or aggregation to response any OLAP queries. In general, tuples from different cuboids are shared if they are known being aggregated from the same set of tuples.

Data Cube Materialization with MapReduce

MapReduce is widely used distributed programming model for various distributed applications. It has been one of research direction using MapReduce for Data Cube Materialization, which utilizes data resides in Distributed File System (DFS) and apply Data Cube Materialization with some MapReduce algorithm.

MapReduceMerge

Yuxiang W., Aibo S. and Junzhou L. proposed MapReduceMerge, a parallel cube construction model based on Google MapReduce and Google File System. They have extended standard Google MapReduce programming primitive as the following to support multiple related heterogeneous dataset in Data Cube materialization.

$$\text{map: } (k1, v1)a \longrightarrow [(k2, v2)]a$$

$$\textit{reduce: } (k2, [v2])a \longrightarrow (k2, [v3])a$$

$$\textit{merge: } ((k2, [v3])a, (k3, [v4])b) \longrightarrow [(k4, v5)]c$$

As it states above, the reduce function in MapReduceMerge produces a list of key/values, which is the input for the additional merge function. The merge function combines output from reduce function from different data collections and result a list of key/values belongs to target materialized dataset.MapReduceMerge employs GFS to segment the data vertically according to the dimensional attributes. That is, a raw data set abcm, where a, b and c aredimensional fields, and m is a measure field, produces three files separately onGFS containing data as *am*, *bm* and *cm*. In this way, it allows to load only relevant data for processing and ignore unnecessary data loads.

Conclusion

The main method for OLAP to provide online responses to analytic queries is to materialize all possible data cubes for a given record. In other words, OLAP directs the analytic queries to the relevant materialized data cube and responds immediately, without executing data aggregations at runtime. However, the data cube materialization is a challenge for both storage and data processing. An n-dimensional dataset generates courses of 2n number of data cubes. In addition, the size of each data cube depends on the cardinality level of each dimension. The CUBE operator initiates the basis for most materialization approaches of the complete data cube while the grid frame receives attention from the direction of materialization of the partial data cube. Among other things, the compression of the data cube, the approximate cube and the parallel data cube are recognizable methods in the materialization of the data cube.

References

[1] S. Chaudhuri, U. Dayal, "An Overview of Data Warehousing and OLAPTechnology", ACM SIGMOD Record, vol 26, pp65-74, 1997

[2] *Readings in Database Systems*, Stonebraker M., Hellerstein J. M, Morgan Kaufmann Publishers, 1998.

[3] *Principles of Distributed Database Systems*, M. T. Özsu, P. Valduriez, Prentice Hall, 1999.

[4] *Data Mining: Concepts and Techniques*, J. Han, M. Kamber, Academic Press, 2001.

[5] *Microsoft SQL Server 2000 Analysis Services Step by Step*, R. Jacobson, Microsoft Press, 2000.

[6] W. Lui, J. Han, "Discovery of General Knowledge in Large Spatial Databases", Proc. of 1993, Far East Workshop on GIS, pp275-289, 1993.

[7] *Spatial Databases with Application to GIS*, P. Rigaux, M. O. Scholl, A. Voisard, Morgan Kaufmann Publishers Inc., 2002.

[8] O. R. Zaiane, Multimedia and Spatial Data Mining, Principles of Knowledge Discovery in Data, University of Alberta, 2002.

[9] N. Stefanovic, J. Han, K. Koperski, "Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes", IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 6, 2000.

[10] V. Harinarayan, A. Rajaraman, J. D. Ullman, "Implementing Data Cubes Efficiently", Proc. 1996, ACM-SIGMOD, Int'l Conf. Management of Data, pp205- 216, 1996.

[11] J. Gray, A. Bosworth, A. Layman, H. Pirahesh, "Data Cube: A Relational Operator Generalizing Group-by, Cross-tab, and Roll-up.", Proc. of the 12th Int. Conf. on Data Engineering, pp152-159, 1996.

[12] S. Agarwal, R. Agrawal, P. M. Deshpande, et. al., "On the Computation of Multidimensional Aggregates", Proc. of the 22nd VLDB Conference, 1996.

[13] Y. Zhao, P. Deshpande, J. F. Naughton, "An Array-Based Algorithm for Simultaneous Multidimensional Aggregates", Proceedings ACM SIGMOD International Conference on Management of Data, pp.159-170, 1997.

[14] A. Laurent, B. Bouchon-Meunier, A. Doucet, et. al., "Fuzzy Data Mining from Multidimensional Databases", Int. Symp. On Computational Intelligence, Studies in Fuzziness and Soft Computing, 54:278-283, 2000.

[15] A. Laurent, "Generating Fuzzy Summaries from Multidimensional Databases", Fourth International Symposium On Intelligent Data Analysis, pp24-33, 2001.

[16] A. Laurent, B. Bouchon-Meunier, A. Doucet, "Towards Fuzzy-OLAP Mining", in Proc. Work. PKDD "Database Support for KDD", pp. 51-62, 2001.

[17] Han J., "OLAP Mining: An Integration of OLAP with Data Mining", Proc. Of the 7th IFIP2.6 Working Conference on Database Semantics (DS-7), pp1-9, 1997.

[18] Kelkar B. "Exploiting Symbiosis between Data Mining and OLAP for Business Insights", DM Direct Newsletter, December 2001.

[19]Vengatesan K., Mahajan S.B., Sanjeevikumar P., Mangrule R., Kala V., Pragadeeswaran (2018) Performance Analysis of Gene Expression Data Using Mann–Whitney U Test. In: Konkani A., Bera R., Paul S. (eds) Advances in Systems, Control and Automation. Lecture Notes in Electrical Engineering, vol 442. Springer, Singapore.