Analysis Of Multidimensional Data With High Dimensionality: Data Access Problems And Possible Solutions

Andrey Borodin[1,*], *Sergey* Mirvoda[1] , and *Sergey* Porshnev[1]

[1] Department of Radio Electronics for Information Systems, Ural Federal University, Russia

Abstract. This article covers the problem of access to multidimensional data with the dimension of 10^4 order. It contains the description of the system and the problem revealed in the process of its building, as well as engineering solutions applied to solve the similar problems. It is also clarified, why such solutions cannot be used for the treatment of the problem concerned. The existing engineering solutions are specified so that it is possible to presuppose the prospects of use of the fragments of such technologies for solving the problem in question.

1 Introduction

Nowadays, multidimensional data structures [1,2] are used in the information-processing systems (IPS) applied in the various fields of human activities, such as the business analysis [3], astronomy [4], geomatics [5], bioinformatics [6], etc. The analysis of their use shows that the rate of multidimensional data access is one of the main factors that influence the operation speed of the IPS. That is why, the development of rapid multidimensional data access algorithms is a highly topical problem. By now, over 30 years [2] have passed after the start of development (the 80-s of the last century) of the algorithms for the general solution of the problem concerned, but despite certain results achieved in this field, no universal automated algorithms adaptable to any specific multidimensional data structure have been developed yet. As a consequence, an IPS cannot be built using the available multidimensional data access technologies without their material updating or creating new algorithms adapted to the peculiarities of the information being collected and processed at a certain production facility.

The authors faced the similar situation when implementing, with a participation of a Russian institution of higher education, a complex project for the development of a high-tech production called "Development of an Automated System for Tracking, Control, Modeling, Analysis, and Optimization of the Full Cycle of Metallurgical Production based on the Creation and Integration of Mathematical Models of Technological, Logistic, and Business Processes of the Production Facility (Automated System for Metallurgical Production, MP AS)" (Code number 2012-218-03-167). MP AS consists of two interrelated modules: the automated information system for collection and analysis of the data of the

* Corresponding author: amborodin@acm.org

production facility (DCA AIS) and the automated information system for the simulation of the technological, logistic, and organizational (business) processes of the production facility (ODM AIS).

The initial information analysis revealed that the structure of data collected and processed by the DCA AIS under development is characterized by a big number of heterogeneous dimensions (the list of measurable process variables, including the online ones, comprises over 7000 names), for which the corresponding data can be retrieved. At the same time, the data characterizing the metallurgical production technological processes are highly integrated and cannot be divided between the different program modules.

One of the possible solutions of the said problem, proved to be reasonable by the experiments undertaken at the stage of the rough design of MP AS with the test data, is based on the use of the fastest up-to-date server hardware and software. Yet, its practical realization would evidently result in the sharp rise of the project cost. Moreover, we cannot be sure that the selected computational capabilities would be sufficient to ensure the required performance of execution of the data warehouse (DW) queries after the long-term operation of the system and accumulation of the big volume of data in the DW. The described situation proves the urgent need to develop the algorithms providing the fast access to heterogeneous multidimensional data of high dimensionality (the data with the total number of dimensions of 10^4 and more).

This article contains the general formulation of the problem concerning the access to multidimensional data with high dimensionality, the results of the analysis of the possibility to apply the available multidimensional data access methods, including the bitmaps, spatial hashing, multiple index connection method, as well as the suggested approach ensuring the fast access to multidimensional data with high dimensionality.

2 Problems of Access to Multidimensional Data with High Dimensionality

Each analytical system is purposed for making certain computations using the respective data. In the process of computations, computation resources are consumed, which are used for solving the two quasi-independent problems: access to the data stored in the DW and handling the extracted data.

This fact can be illustrated by the simple examples given below.

Example 1. There is a Microsoft Excel table containing the columns respectively called *A*, *B*, and *C*. The C column contains values calculated by the formula: $C = A + B$. Thus, the operation can be presented as an abstract syntax tree that consists of one node corresponding to the operation of addition (data handling), and the process of obtaining values from the specific cells of *A* and *B* columns can be considered as the data access.

Example 2. Building the three-dimensional scene on the monitor screen. In this case, the handling of data (mostly, computation of matrix products) is the most costly operation in terms of the use of computation resources. At the same time, the data retrieval requires much less computation resources than the data handling, as the volume of the initial data used in the computations is comparably small.

Example 3. Finding the way in the meeting graph of a large social network linking two and/or more users. In this case, the computations are made with a minimum amount of arithmetic or conditional operations, but due to the big volume of initial data and (typically) the use of the distributed data storage method, significant computation resources are required to access such data.

Historically, solving the problems of the specialized access to multidimensional data has been closely connected with the memory hierarchy. The term “hierarchy” itself also has a historical nature. In general, this is about the way of ranking the types of computer

memories: from the fast, size-limited, and expensive memory (processor registers, cash, RAM) to the slow and cheap memory (hard disk systems, reservation systems, cloud storages). The key aspect of most multidimensional data access methods is lowering the number of queries to the slow memory due to the efficient layout of data in the fast memory.

It is characteristic of the multidimensional data access that the data search (a query) usually refers to several different data properties (different dimensions) at the same time. In this case, the list of the data properties used in the search query is known in advance. However, their number is usually so large, that it is almost impossible to build all one-dimensional search structures for all dimension combinations. For example, a query (that should be efficient in terms of the resource utilization) to a multidimensional data structure containing the anthropometric information about a large number of people, purposed to find the people having the nose length, the height, the birth date falling within the specified intervals. Here, the intervals setting the required values for the nose length, the height, and the birth date serve as the search criteria, whereas the information fields containing the values of the nose lengths, heights, and birth dates of the people included in the database (DB), are the active dimensions of the search query. A database may have a number of different dimensions, but a multidimensional data structure should ensure the equal efficiency of all queries made to it, regardless of the combination of available active dimensions used in the respective query.

Below are given the examples of information systems, where multidimensional information is being processed.

Example 4. Geographic information systems. Although these IPS traditionally operate with 3 to 4 dimensions (3 spatial coordinates and the time), they are also referred to the class of information systems with the multidimensional data processing. This is conditioned by the high variability of the queries used in such systems, e.g. "the search of the houses within the given rectangle", "the search of the gas stations nearest to the car location", or "the search of the best route with three or more changes", etc.

Example 5. Business analysis systems. Such IPS handle the data using dozens of search dimensions [7]. For example, a query of a business analysis system may look as follows: calculate the total income from all operations performed under the certain group of tariffs, within the certain date ranges, in the certain district of the city, for the subscribers having the connection to the digital TV and the broadband Internet access.

Today, there exist a great number of various data structures; some of them are described in [8]. Most of today's data structures represent the subset of the so called generalized index search tree (GiST) [9], formed by way of division of the whole data space into the hierarchically nested groups being the nodes of the tree. The grouping of objects in the indexed data space is performed so that the number of groups affected by the most probable data queries is reduced to a minimum.

A data access structure is selected based on the structure of the data and the query. The so called search query efficiency models are used to substantiate the selection of a specific balanced tree structure in quantitative terms. For example, in [10] it is offered to estimate the efficiency of the spatial indices used for making queries to the business analysis multidimensional data using the following model:

$$p(|q|,|s|)=\frac{|s|+|q|}{(1-|q|)(1-|s|)},$$

$$W(0,|s|)=|s|,$$

$$W(x+1,|s|)=\frac{\sqrt[D]{F}-1}{\sqrt[D]{N(x)}}+W(x,|s|).$$

$$DA=1+\sum_{x=1}^{[\log_F N]}\frac{N}{F^x}\prod_{j=1}^{D}p(W(x,|s_j|),|q_j|),$$

Fig. 1. The Model for making queries to the business analysis multidimensional data

where *DA* (disk access) is a number of the analyzed nodes of the tree data index;
N is a number of indexed records;
F (fanout) is a branching ratio of the tree index;
p(s,q) is a probability of intersection of *s* and q ranges;
W(x,s) is an average data range for one dimension grouped *x* times;
D is an indexed data dimensionality;
s is an average range occupied by one element of the initial data.

Moreover, as specified in [10], in the general case when we deal with the business analysis data with the number of dimensions of 10^1 order, the complexity of computation of the fixed result query (the number of data meeting the query criteria) — the ratio between *DA* and the size of the resulting sample has the complexity O(*N* log *N)*. However, at larger *D* values (e.g, those of 10^4 order), $\sqrt[D]{F}\approx 1$ and $W(x+1,|s|)\approx W(x,|s|)$ it means that the interlevel data grouping is inefficient due to the reduction of range of the analyzed tree nodes when going down the index tree. As a result, the complexity of the fixed result query computation turns out to be O(*N*).

3 Problems of Bitmap Indexing of Multidimensional Data with High Dimensionality

The bitmap indexing method [11] (the bitmap method), as compared to the method of balanced trees, makes it possible to realize a simpler, as regards the computations, technology of organization of the quick search of data by multidimensional conditions. This technology is based on the ideas used in the binary computation facilities. Under the bitmap indexing method, each possible element of the data dimension is correlated with a bitmap, which is filled in with zeroes or ones depending on the consistency or inconsistency of values of the fields of all DB rows to the query field values. The sequence of the bitmap completion repeats the sequence of the DB data fields retrieved for the query. The search of the rows meeting the conditions of the conjunctive query is performed by the bitmaps multiplication. After that, the ones remaining in the obtained product correspond to the numbers of the rows meeting a condition of the multidimensional query. The technology is proved to be efficient; it is utilized in most up-to-date DBMS.

The technology allows to efficiently determine the identifying information of the data rows. However, the data access problem has not been yet fully solved in it, as after finding the rows meeting the query conditions, it is necessary to obtain the values of the attributes of the retrieved rows using the identifying information (lookup operation). In terms of the

DBMS terminology, this means that a bitmap is a noncluster index. For the physical retrieval of multidimensional data, a Gilbert codes arrangement is used, e.g. in Microsoft Analysis Services, as follows from the comments posted by one of the authors of this method in his blog [12]. At the same time, the study [13] contains the examples, where such method appears to be inefficient for the work with the high dimensional data.

The problem arising when using lookup operations for the computation of the multidimensional data query can be illustrated by the following example. Let us assume that we have N rows with D dimensions, each of which possesses on average M possible values. This information should be arranged in the data blocks of F rows, so that each query with one condition (the total number of possible variants — $D \cdot M$) would require the use of a minimum possible number of information blocks DA, which should be referred to when obtaining access to the data meeting the query conditions. However, when $D>>F$, it is impossible to minimize DA, if D dimensions include no interrelated ones; in other words, the grouping of the data rows under one attribute results in the serious deviations of another attribute.

It should be noted, that most of available spatial indexing methods are based, inter alia, on the interrelation between the dimensions. Spatial indices can use certain local dependencies detected in the certain data subsets considered as nonlinear ones. We believe that nonlinear relations (e.g., between the groups of dimensions) can also be effectively used when applying the bitmap indexing method for the high dimension data access.

4 Spatial Hashing Method

The locality-sensitive hashing method (LSH) [6]) is worth mentioning, as it is successfully applied in the different fields of information technologies [14] dealing with multidimensional data of high dimensionality. The key idea of the method implies that a hash code coinciding with high probability is generated for the spatial points located close to each other.

There are three basic ways of using LSH method to access the data:

1. To generate a hash code for the query and the data row, so that one hash code corresponds to one query. All records meeting the query conditions should have the same hash code.
2. To generate a hash code, so that a set (in most cases — a multidimensional range (parallellotope)) of hash codes corresponds to one query. Accordingly, the records meeting the conditions of such query should have one of the possible hash codes, i.e. a point of multidimensional space belonging to the parallelotope.
3. To generate a hash code, so that one hash code corresponds to one query. In such a case, the considered hash code is assigned to most records meeting the conditions of such multidimensional query.

The first of the described data access methods is actually a one-dimensional method, as different combinations of dimensions would require different hash functions. The second method, in case of the growth in the number of dimensions, is prone to the ''combinatorial explosion" of the parallelotope volume. Consequently, their efficiency is below the efficiency of other available methods of multidimensional data access.

The third method cannot ensure the correct result of the query execution.

5 Method of Multidimensional Indices Junction

One of the most popular engineering practices used in relational databases when solving this problem is building a specific index for each dimension. During the execution

of a multidimensional query, RDBMS joins the results of filtration of the separate conditions by the corresponding indices.

The simplicity of programming and reliability (in terms of reduction of probability of logical errors in the algorithms for the multidimensional query computation) are the obvious advantages of the suggested structure of a cluster index of a one-table multidimensional data warehouse. It is also noteworthy that the efficiency of the said index is independent of the number of dimensions in the indexed data. At the same time, it is probable that such indicator would be rather dependent on the number of dimensions stated in the query conditions. In fact, the data access system would need to compute all samples for the separate query conditions, each of which can appear much bigger than the sample corresponding to the conjunction of the query conditions.

6 Conclusion

The above analysis of the existing methods for the organization of the data access for the computation of multidimensional queries in the systems with a number of independent data dimensions of 10^4 order has revealed that neither of them can be deemed a universal instrument for solving the considered problem. Thus, there is a real need in the development of new problem-solving approaches. We deem it promising to apply the method based on the use of the hybrid technology integrating one or more of the above mentioned multidimensional data access methods. Such technologies have already proved their efficiency in solving the problems of access to multidimensional data with a number of dimensions of 10^0-10^2 order— for example, a so called BR-tree [15] with a Bloom bit filter inserted in its nodes [16].

The advantage of the bitmaps, which can be used in the hybrid method, is the absence of the explicit dependence of the query computation efficiency on the number of the indexed data dimensions, that is why they are used when solving the problems of 10^4-dimensional data indexing. Above that, when spatial indices are used, it is possible to use local nonlinear dependencies between the dimensions. We believe that the combination of these technologies would allow to create the system for analyzing the multidimensional data with high dimensionality, where the time needed to obtain the computation results would be measured in dozens of milliseconds rather than dozens of minutes.

References

1. V. Gaede. Multidimensional Access Methods / Gaede V., Gunther O. // ACM Computer Surveys. 1998. Vol.30. No. 2. P. 170-231.
2. D. Greene. An Implementation and Performance Analysis of Spatial Data Access Methods/ Greene, D. // In Proceedings of the Fifth IEEE International Conference on Data Engineering. 1989. P. 606-615.
3. A. M. Borodin, S. V. Porshnev, M. A. Sidorov. Application of Spatial Indices for the Processing of Analytical Retrievals and Aggregation of Multidimensional Data in IAS. Tomsk Polytechnical University Publishing. 2008 No 5, Pp. 64-86.
4. M. Frialis. Data Management and Mining in Astrophysical Databases // PhD thesis, Univ. of Udine, Italy. 2005.
5. K. T. Chang. Introduction to Geographical Information Systems/ K. T. Chang // New York: McGraw Hill.-2008. P. 184.
6. A. Andoni. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions / Alexandr Andoni and Piotr Indyk // Communications of the ACM. Vol. 51. No. 1. 2008. P. 117-122.

7. A. M. Borodin, S. V. Porshnev. Comparative Analysis of the Possibilities and the Rate of Processing of Multidimensional Data by the Business Analysis Software Based on the Main Memory Indexing Structures. Scientific and Technical News of SPbGU. Series ''Computer Science, Telecommunication, Management" -2010. –No. 1. Pp. 99-102.
8. V. K. Gulakov, A. O. Trubakov. Multidimensional Data Structures. 2010. Bryansk: BSU Publishing, P. 387.
9. J. Hellerstein. Generalized Search Trees for Database Systems /J. Hellerstein, J.Naughton, A. Pfeffer// Proc. 21st Int'l Conf. on Very Large Data Bases. Zurich. September, 1995. P. 562-573.
10. A. M. Borodin, S. V. Porshnev. Analytical Methods for Estimating the Efficiency of Spatial Indices Application in OLAP Systems. [Text]. Scientific and Technical News of SPbGU. Series ''Computer Science, Telecommunication, Management" -2011. Pp. 93-100.
11. T. Johnson. Performance Measurements of Compressed Bitmap Indices / Johnson T.// - Proceedings of 25th International Conference on Very Large Data Bases. September 7-10, 1999. P. 278-289.
12. M. Posumanskiy. Chronicle Number 9 "We Are Great Minds". [Digital resource] // URL : http://web.archive.org/web/20040306084024/http://www.mosha.com/XRONIKI/win-xronika9.html (access date 10/13/2013)
13. A. M. Borodin, S. V. Porshnev. Algorithms for the Quick Access to Multidimensional Data in OLAP Systems. Saarbrucken: LAP Lambert Academic Publishing. 2012. P. 176
14. A. Rajaraman. Mining of Massive Datasets, Ch. 3. /A. Rajaraman, J. Ullman // Stanford University, California, 2010. P. 326.
15. Yu Hua. BR-Tree: A Scalable Prototype for Supporting Multiple Queries of Multidimensional Data /Yu Hua , Bin Xiao, Jianping Wang // Computers, IEEE Transactions on. Vol. 58. Issue 12. 2009. P. 1585-1598.
16. B. H. Bloom. Space/time trade-offs in hash coding with allowable errors/ Burton H. Bloom // Communications of the ACM T. 13 (7). 1970. P. 422-426.