Clustering method based on the analysis of the access request stream in object-oriented databases

K. TOMASZEWSKI
ktomaszewski@wat.edu.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

Recent studies on modern database management systems consider object-oriented databases as a possible significant extension of the modern database functionalities. However, new functionalities bring an increased processing complexity. This may also result in an increased demand for computing resources and the deterioration of database performance. In this article, the use of clustering methods has been described as one of performance techniques in object-oriented databases. The article includes an introduction to the popular clustering methods discussed so far. Afterwards, a new method has been introduced in order to analyse the access request stream as the basis for a new clustering approach in OODB. Graph techniques are discussed as the basic elements of the familiar clustering methods and their adaptation to the aforementioned new approach. This article also describes limitations of the existing methods and the possible impact on the new technique. Finally, selected algorithm variants are proposed for the new clustering method to improve performance of object-oriented databases.

Keywords: clustering, object-oriented database, OODB.

1. Introduction

Research and development of databases is motivated by the need for better support of design and maintenance of modern information systems. Object-oriented databases are one of the significant threads of this research as they bring a number of new features to modern database solutions. Unified meaning of an object-oriented database system is still open to discussion. However, the general principle is to combine the features of object-oriented programming languages, such as data model abstractions, with database features, such as the possibility to persist the data. The object-oriented data model extends the capabilities of the relational data model, i.e. with the use of complex data attributes. Simple column values, known from relational databases, can be extended by simple, as well as complex object attributes, such as sets, lists or references to another object instances. One of the most significant innovations is the introduction of object identity as a way of accessing the specifically desired data instance with the use of reference attributes.

2. Performance of object-oriented databases

Extended capabilities of object-oriented databases increase, however, the general complexity of database management systems. This may also result in an increased demand for computing resources required for processing database operations. Deterioration of general database performance may involve an increased time of serving the requests incoming from database users. This factor motivates the design and development of a wide variety of performance techniques for object-oriented database management systems.

In the bibliography, we can find numerous descriptions of research aimed at improving the effectiveness of the aforementioned class of systems.

Selected aspects of general performance research were described in [1]. They include clustering, partitioning, replication of data, as well as concurrent processing or indexing techniques in data access processes.

3. Data Clustering

Data clustering techniques have been selected from the above-mentioned performance research as a subject for further analysis. Clustering itself includes a wide scope of knowledge that preceded the creation of first computers. Modern definitions of clustering mainly relate to the issues of cluster analysis and identification of thematically interrelated groups of entities. Cluster analysis is based, however, on statistical classification algorithms which assign statistical observations to the identified classes, based on the characteristics of such observations.

The specific nature of database management systems provides an opportunity to use the information from the cluster analysis, based on fact that single objects within a database are usually grouped together and stored in larger, separate logical units. This phenomenon has been implemented in two popular database organization models.

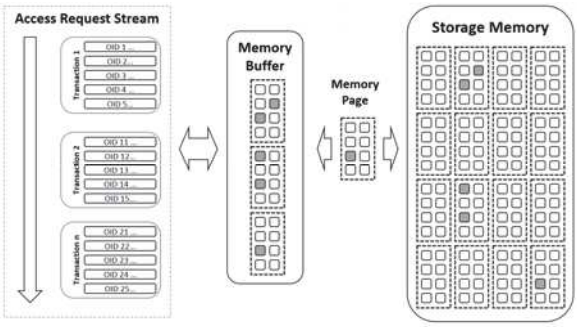

Fig. 1. Buffer memory model

The first one is a model of a centralized database management system. It is based on a single database unit, serving all incoming access requests. In this model it is assumed to serve access requests with the use of buffer memory (cache, RAM), as well as storage memory (HDD) of the database system. Objects stored in the storage memory are grouped in memory pages as a larger logical memory units (figure 1). Memory pages are used as the basic data exchange units between the buffer and the storage memory. Serving of incoming access requests involves loading the whole page, which contains the referenced object, into the buffer. If the subsequent access requests refer to the objects, stored within the same page, they will be served with the use of buffer only (no further HDD read will be necessary).

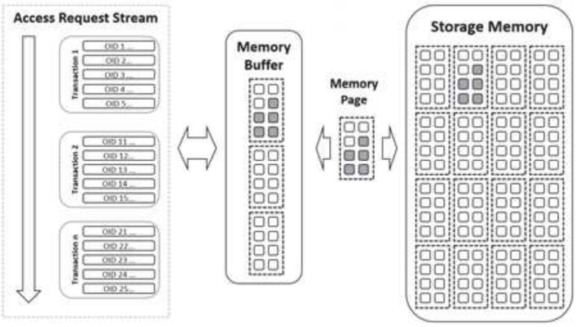

Fig. 2. Related objects stored together in logical memory units

The aim of clustering, in general, is based on the assumption that the subsequent incoming access requests will refer to objects with identified similarities, in terms of a particular similarity criterion. The goal is to provide an effective similarity criterion, embedded in a complete clustering approach, in order to identify groups and store them in correct logical units (figure 2).

The second model, which involves grouping data into bigger logical units is a distributed database model. In such models, data objects are distributed among a number of database nodes, operating in a shared network structure. An access request, incoming to a particular database node, will be served with the use of data stored locally and if necessary, additional requests to the remote nodes will follow.

Effective strategy for object location among the database nodes will result in serving the subsequent access request with the use of local resources, while reducing the number of remote accesses. The second model is a higher-level abstraction, which includes also a previous buffered-memory model for every single database node.

4. Existing clustering techniques

Recent work related to similarity evaluation has brought a comprehensive variety of techniques for analysing entity properties.

One of the general approaches is to define a measure and evaluate entity similarity based on selected entity attributes [2], [6].

Examples of more specific object-oriented methods assess the object similarity from the data model description, e.g. inheritance dependencies [3], [4], [5] and by analysing associations between the actual object instances [6], [7]. It is also possible to examine a set of predicates fulfilled by particular data instances [9]. These methods can operate in the data model domain, as well as data instance domain. Other examples of the method are able to examine the popularity of data entities in order to perform effective clustering (so called HOT-COLD techniques) [8].

Most of the existing clustering techniques can be described with the use of a general clustering algorithm presented below.

First, essential entity features (attributes) are selected, catering significant information for further similarity deduction. Examples of such features include values of chosen attributes, as well as a mutual object-oriented class inheritance hierarchy or common sets of fulfilled query predicates.

Next, the representation format of the selected features is adapted to serve as an input for a chosen clustering tool. At the same time, a similarity measure needs to be defined. This measure is usually specific to a particular clustering tool. With a defined similarity measure, the chosen clustering tool may be used to identify clusters in the object set. Examples of popular clustering tools include such methods as the k-means method [10] or hierarchical clustering [11]. Outcome results can be then analysed and information about identified clusters may be used for the purpose of reorganisation of the database.

5. Access request stream analysis

In addition to the existing ones, a new clustering technique has been introduced.

Unlike the previously described techniques, the new technique does not involve an analysis of internal object features. It is based, however, on the observation and analysis of the access request stream incoming to the database system. The goal of the analysis is to identify groups of objects which are accessed in relatively short time intervals. Proximity of access requests in time is then used as a grouping criterion. In the implementation of the method, recorded information regarding the last access time for each object instance is used. Based on such information, various groups of related objects are identified (figure 3).

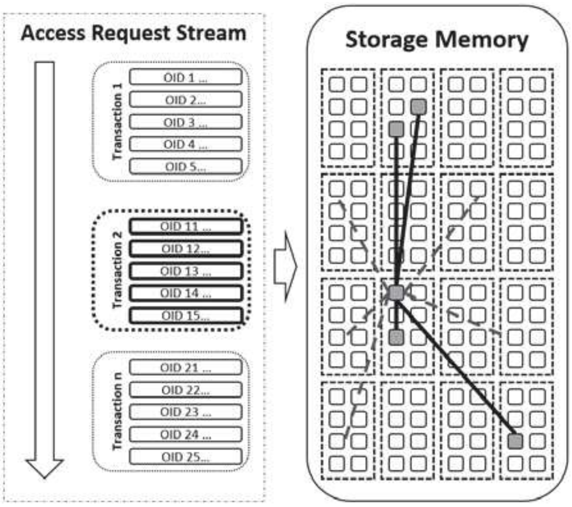

Fig. 3. Identification of object group in access request stream

Identified object groups are then placed on the memory pages (figure 4).

Observation of the access request stream is the only source for the object grouping deduction. Low-level input data simplifies the implementation method, as compared to other popular clustering methods which are based on complex analysis of data models or analysis of associations between object instances.

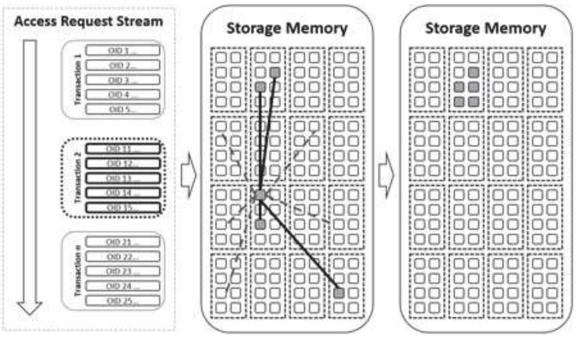

Fig. 4. Data reorganization

If the objects referenced by the subsequent access request are placed on the same memory page, there is no need to load a new page into the buffer. As a result, the overall page swapping process will be reduced in the database management system processing.

6. The use of CFNG graph

For the purposes of the approach described above, a clustering tool needs to be provided in order to perform actual clustering into proper groups. From among a variety of clustering tools operating in metric space, the *Colored Farthest*

Neighbour Graph method (CFNG) has been selected, as it introduces some additional features described in chapter 9 of this paper.
The CFNG graph reflects the relation of the farthest neighbourhood according to a chosen distance measure. The *Colored Farthest Neighbour Graph* method has been originally described by A. Hausner in [12]. It originates from the *Shared Farthest Neighbour Graph* method, described by S. Rovetta and F. Masulli [13].

At first, a similarity measure should be defined or more precisely, a distance measure between objects being clustered. As the approach involves grouping objects with the close requests occurrences, the intuitive choice is to determine the distance measure $d(O_m, O_n)$ as a difference of time values between access requests and the objects O_m and O_n in the linear time domain (figure 5).

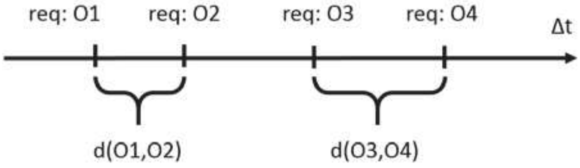

Fig 5. Distance measure in object access request stream

For each object in the analysed set, the distance to any other objects is evaluated. Based on the measures, the *Farthest Neighbour Graph* (FNG) is created then. In the graph, the vertices represent particular objects. Each edge connects the object to the one with the maximum distance value. This means that at least one of the objects connected by the edge is the farthest neighbour of the other, according to the defined distance measure (figure 6A). Afterwards, the resulting graph is subjected to the colouring process. In [12] A. Hausner shows that the FNG graph does not contain any cycles, thus it is always a tree structure (figure 6B). As a result, a two-colour model is always present in the graph (figure 6C).

By colouring, each node is given a label which assigns it to one of the sub-clusters. As a result, we arrive at a two-colour graph (figure 6D). Each iteration of the algorithm divides the processed set binary into two subsets.

The stop-condition of the algorithm can be determined in various ways. The number of iterations may depend on the expected properties of the final clusters or just on the expected number of the clusters.

It is also possible to construct a full cluster hierarchy, by repeating the binary division until the result of N-number of single-element subsets of the N-element initial set.

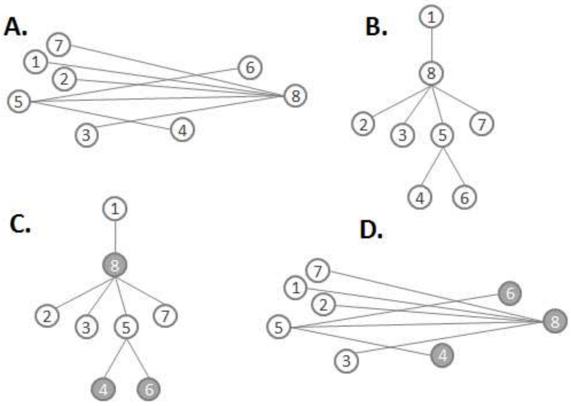

Fig. 6. The CFNG Method: A. Setting the edges; B. Tree structure; C. Graph colouring; D. Identified clusters

Usually, size features of the objects and the capacity of the cluster containers are determined. In this case, the clustering method can be executed until every outcome cluster can be stored in a single cluster container.

7. CFNG method limitations

Several situations can be described where the CFNG method can result in poor quality clustering. Although it is certain that the steps of the algorithm can be properly executed in each case, it cannot be ruled out that the algorithm itself will reflect the natural properties of the observed clusters in an incorrect manner. The literature on the subject [12] describes a situation susceptible to incorrect algorithm outcome. It results from the binary-division feature of the method. In each algorithm iteration, every element of the examined set is assigned to one of two sub-sets. These sub-sets reflect the natural “poles” of the set (based on the most remote points). If the analysed set contains a group of elements located near the middle point between the poles, the algorithm will break down the middle group by assigning each element of it to one of the two border clusters (figure 7).

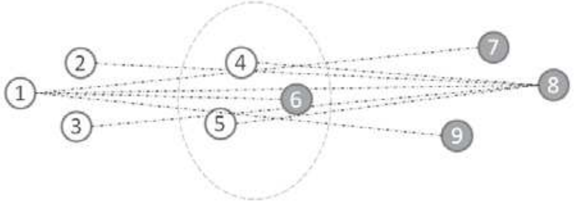

Fig. 7. Hasty split in the CFNG method

Such division is called *hasty split* and is also possible for other clustering methods. The results of low quality hierarchical divisions have been described in detail in the literature on the subject. A number of repair, preventive and mixed techniques have been also proposed to counteract low quality divisions.

8. Simplification of the algorithm

Significant simplification can be observed when CFNG method operates in the linear metric space. In each iterations it is possible to distinguish two border objects: *A* and *B* with the lowest and the highest position value in linear space. Additionally, the mid-point *m* can be also distinguished in the space between the extreme points dividing it into two equal areas (figure 8). It is possible then to assign every object to one of the sub-clusters using simple inequality. All the objects with the position value lower than the mid-point will be assigned to one cluster with the extreme maximum object as the farthest neighbour. By analogy, all the objects with the position value higher than the mid-point will be assigned to the second cluster with the extreme minimum object as the farthest neighbour.

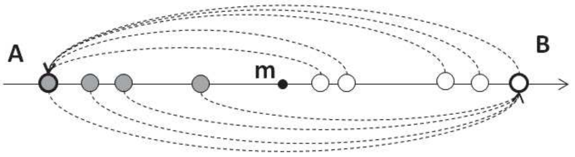

Fig. 8. Simplification of the CFNG method in linear space

Simplification however does not relieve the method of the previously described "hasty split" among the clusters located near the centre of the set in the space of time.

9. Algorithm modifications

Modular structure of the clustering algorithm provides an opportunity to replace the distance measure between the objects, while using a CFNG graph as a clustering tool. A significant feature of the CFNG is that it enables the clustering of objects without knowledge of their location in metric space. The distance measure may be then defined as an abstract value resulting from the other object model features. An example of an abstract distance measure, which may be used as a substitute, is the aforementioned feature of affiliation of object classes to common inheritance hierarchies [3], [4], [5]. Furthermore, other distance measures, for example vector methods – taking into account a greater number of feature dimensions – can be considered as replacement candidates.

Moreover, the possibility to present the location of the object in the linear time domain creates an opportunity to use such information as an input data for other clustering methods. The CFNG graph construction method can be then easily replaced and compared to other methods during actual evaluation experiments.

10. Summary

This article refers to the growth in complexity of processing in object-oriented databases. This growth is related to the introduction of new features and functionalities to modern database systems provided by object-oriented databases. Main innovations are mostly related to the object-oriented data model. The complexity growth affects, in an adverse manner, the efficiency of serving access requests incoming to the object-oriented database. This article describes the search and evaluation of efficiency techniques designed for this class of systems. The following parts elaborate on clustering as one of familiar efficiency techniques used in database management systems. The article contains a concise introduction to popular clustering methods. It also presents a new clustering method, based on the access request stream analysis for object-oriented databases. In addition, the study discusses special features of coloured graphs as a basic tool of the proposed clustering method. Along with the detailed presentation on those features, the paper proposes simplifications and various implementation models of the clustering algorithm. The use of the proposed method in object-oriented databases is aimed at improving efficiency features of those systems.

11. Bibliography

[1] K. Tomaszewski, „Obiektowe bazy danych – wybrane kierunki rozwoju", *Biuletyn Instytutu Systemów Informatycznych*, No. 10, 53–61 (2012).

[2] J. Ahn, H. Song, H. Kim, "Index set: A practical indexing scheme for object database systems", *Data & Knowledge Engineering*, No. 33, 199–217 (2000).

[3] Y. Huang, J. Chen, "The study of indexing techniques on object oriented databases", *Information Sciences*, No. 130, 109–131 (2000).

[4] W. Cho, W. Han, J. Lee, I. Song, K. Whang, "2D-CHI: A Tunable Two-Dimensional Class Hierarchy Index for Object-Oriented Databases", *Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International*, 598–607, 2000.

[5] W. Cho, W. Han, J. Lee, I. Song, K. Whang, "A tunable class hierarchy index for object-oriented databases using a multidimensional index structure", *Information and Software Technology*, No. 43, 309–323 (2001).

[6] K. Karlapalem, Q. Li, "A Framework for Class Partitioning in Object-Oriented Databases", *Distributed and Parallel Databases*, No. 8, 333–366 (2000).

[7] L. Bellatreche, K. Karlapalem, A. Simonet, "Algorithms and Support for Horizontal Class Partitioning in Object-Oriented Databases", *Distributed and Parallel Databases Journal*, No. 8, 155–179 (2000).

[8] S. Blackburn, Z. He, R. Lai, A. Marquez, "Opportunistic prioritised clustering framework for improving OODBMS performance", *Journal of Systems and Software*, No. 80, 371–387 (2007).

[9] A. Campan, O. Cret, A. Darabant, "Hierarchical Clustering in Object Oriented Data Models with Complex Class Relationships", *Intelligent Engineering Systems. 8th IEEE International Conference*, 307–312, 2004.

[10] A. Darabant, "Semi supervised learning techniques: k-means clustering in OODB Fragmentation", *Computational Cybernetics. Second IEEE International Conference IEEE*, 333–338, 2004.

[11] F. Murtagh, "A Survey of Recent Advances in Hierarchical Clustering Algorithms", *The Computer Journal*, No. 26, 354–359 (1983).

[12] A. Hausner, "A new clustering algorithm for coordinate-free data", *Pattern Recognition*, Nr 43, 1306–1319 (2010).

[13] S. Rovetta, F. Masulli, "Shared farthest neighbor approach to clustering of high dimensionality, low cardinality data", *Pattern Recognition*, No. 39, 2415–2425 (2006).

Metoda klasteryzacji uwzględniająca charakterystyki strumieni żądań dostępu do danych w obiektowych bazach danych

K. TOMASZEWSKI

Rozwój obiektowych baz danych związany jest z rozszerzeniem możliwości współczesnych systemów bazodanowych. Nowe funkcjonalności związane są jednak ze wzrostem złożoności przetwarzania oraz mogą wpływać na pogorszenie wydajności baz danych. W artykule tym omówione zostało zastosowanie klasteryzacji jako jednej z technik poprawy wydajności w obiektowych bazach danych. Artykuł zawiera wprowadzenie do popularnych metod klasteryzacji omawianych dotychczas. Następnie opisana została metoda analizy strumienia żądań dostępu do danych jako podstawa nowej techniki klasteryzacji w OODB. Omówione zostały również właściwości kolorowanych grafów oraz ich zastosowanie w nowej metodzie. Wraz z opisem nowej metody przedstawione zostało możliwe uproszczenie technik grafowych, jak również wybrane warianty modyfikacji algorytmu metody klasteryzacji.

Słowa kluczowe: klasteryzacja, obiektowe bazy danych, OODB.