Optimization of Data Sets Allocation in the Distributed Databases

Sergiy V. Lazdyn, Alexander O. Telyatnikov

Department of Automated Control Systems
The Donetsk National Technical University
58, Artema Str., Donetsk, 83000, Ukraine
slazd@ukr.net
Alexander.Telyatnikov@gmail.com

Abstract: This paper deals with the new method of solving the problem of data set allocation optimization problem in the distributed database systems. The method is based on the combined use of object model of the distributed database and modified genetic algorithm. The criterion for optimum is defined by the minimum total average time required to process queries and disseminate updates. The experimental research of the developed method has proved that the productivity of the information system can be increased by 30% due to optimization of the dataset allocation without the need for additional expenses on hardware modernization.

1 Introduction

The distributed databases (DDB) are used when designing the information systems for large-scale enterprise. The data sets in DDB are arranged in the great number of nodes by means of fragmentation and replication. Such systems functioning can be maintained by most modern DBMS (IBM DB2, Oracle, MS SQL Server, Ingress and other) with tools for the DDB design and back up.

DDB's system is a complex dynamic object, which carries out a great number of queries to the distributed data fragments, updates a great number of the fragments copies arranged on different nodes of computer network. The DDB productivity is affected not only by the hardware parameters (servers, communication channels) but also by the rational distribution of data in the system. Therefore, the problem of the DDB optimization aimed at improving their high efficiency should be resolved both by the development of new information systems and modernization of the existing systems.

The previous papers [Ga00, Ts00] concern DDB design that had been carried out by means of analytical models, while optimization was achieved due to mathematical programming. We'd like to point out the fact that disadvantage of these projects is that analytical models do not allow to take into account the dynamics of queries processing

and updates dissemination in DDB. Another point out of its favour is that mathematical programming methods are difficult to apply for the optimization of information systems with a large number of nodes and data fragments. The paper under consideration [Co00] suggests carrying out optimization of data fragments allocation in DDB using genetic algorithms in combination with an analytical model. It should be also stressed that this approach doesn't take into consideration data replication and delay in queries processing caused by the heavy load of a network and the information system nodes.

This paper dwells upon a new method of optimization of data sets allocation in DDB for big information systems based on the use of the DDB object model in combination with genetic algorithms (GA) [LT00, LT01, Te00].

2 Problem definition of the DDB optimization

The time necessary for system reaction is the major indicator of DDB operation speed at which the queries are processed and updates are disseminated. This is an interval from the moment the query or the update is initiated up to the moment the system signals that the queries have been processed and updates have been spread. This time depends on the productivity of data processing nodes, carrying capacity of data transmission channels, volume of transmitted data and on data allocation on the DDB nodes. Therefore, the total average time required for queries been processed and updates been spread during the set time period is suggested to be the criterion for DDB efficiency estimation. [LT00].

The general scheme of DDB is composed of great number of data fragments $F = \left\{ f_i \mid i = \overline{1,n} \right\}$, where n is the number of data fragments in the system; which in their turn are allocated on a great number of nodes by means of fragmentation and replication $Y = \left\{ y_j \mid j = \overline{1,m} \right\}$, the nodes are connected with a number of data transmission channels $C = \left\{ c_k \mid k = \overline{1,N_c} \right\}$. a great number of applications is carried out on the nodes $A = \left\{ a_r \mid r = \overline{1,N_a} \right\}$ applications initiate queries processing and updates spreading. The scheme of data allocation in DDB is determined by a matrix $X = \{ x_{ij}, i \in [1,n], j \in [1,m] \}$, the elements of which x_{ij} take on values: 1, if the copy of data fragment *i* and 0 is allocated on the node *j*, in other case.

Provided the DDB functions together with the scheme of data allocation, represented by matrix X, a great number of queries $Q = \left\{ q_s \mid s = \overline{1,N_q} \right\}$ and a great number of updates $U = \left\{ u_e \mid e = \overline{1,N_u} \right\}$ is generated for the elements the functions of which are defined $t'\left(q_s, X\right)$ as time required for queries processing $q_i \in Q$, $s = \overline{1,N_q}$ and $t''\left(u_e, X\right)$ as time needed for updates spreading, $u_e \in U$. $e = \overline{1,N_u}$

The task for the DDB optimization can be formulated in the following way: it is necessary to find the scheme of data sets allocation on the nodes of the information system, which would bring the total average time required for queries been processed and updates been spread T to the minimum:

$$T(Q,U,X)=\frac{1}{N_q}\sum_{s=1}^{N_q}t'(q_s,X)+\frac{1}{N_u}\sum_{e=1}^{N_u}t''(u_e,X)\to\min \tag{1}$$

The following limitations must be taken into account when calculating the criterion (1) values:

1. DDB must have at least one copy of every data fragment:

$$\sum_{j=1}^{m}x_{ij}\geq 1,\quad \left(i=\overline{1,n}\right) \tag{2}$$

where x_{ij} are elements of data allocation matrix $X=\{x_{ij}, i\in[1,n], j\in[1,m]\}$.

2. The overall amount of information kept on a node must not exceed common disk space of this node:

$$\sum_{i=1}^{n}L_i\cdot x_{ij}\leq D_j,\quad (j=\overline{1,m}) \tag{3}$$

where L_i is volume of the data fragment, $i\in[1,n]$; D_j is disk space occupied by node j, $j\in[1,m]$;

3. Maximal time required for queries processing must not exceed the set maximum value:

$$\max_{\forall q_i\in Q}\{t'(q_i,X)\}\leq t_{lim} \tag{4}$$

where t_{lim} – critical time(the permissible maximum) required for queries processing.

This task is referred to the class of combinatorial tasks. The number of possible combinations of data allocation on computer network nodes is determined by expression:

$$M=\left(2^m-1\right)^n, \tag{5}$$

where m is the number of the DDB nodes; n is the number of data fragments which are to be allocated on the network nodes.

Due to the complexity of computing and its largeness it is impossible to resolve the problem of optimization of data allocation in DDB by means of classical methods. Therefore, we suggest using genetic algorithms to find optimum scheme for data allocation in information system. Besides that , we suggest calculating the time required for queries processing and updates spreading with the help of the DDB object model [LT01].

3 DDB object-oriented model

The object-oriented approach that allows to describe different complex systems in operation has been chosen for construction of the DDB model. Different DDB structures have been analyzed and the following typical DDB components have been defined: node, data transmission channel, application, query, data set. To construct the models of these typical DDB components we developed relevant classes of objects. To describe features and methods of typical components and to determine the way they interact and behave the Unified Modeling Language (UML) was used. Let's consider the object models of the typical DDB components [Te00].

The "Node Class" simulates work of the DDB node engaged in queries processing and updates spreading. This class can be characterized by: productivity, overall volume of disk space, state, queue for processing. Methods of the class are as follows: query processing, forming a queue for query processing, node release.

The "Channel Class" simulates the work of data transmission channels related to queries processing and updates spreading. The class is characterized by the following features: carrying capacity, state, traffic, and queue for transmission. The methods of class are as follows: data transmission, forming the queue for data transmission, channel release.

The "Application Class" simulates work of applications that is carried out on the DDB nodes. The applications initiate queries and updates. The class possesses the following features: processing nodes; initiated queries; expected values and dispersions of intervals between the starts of applications; times of their execution, intervals between the moments of queries initiation. Methods of class: application start; completion of application; initiation of query.

The "Query Class" simulates the query (data reading or updating) in DDB. Its basic features are: initiated subqueries; expected value and dispersion of time interval value between the moments of initiation of subqueries; query volume; complexity and response volume. Methods of the class: query start; forming the queue for transmission; transmission; forming the queue for processing; processing; queuing up response in the transmission process; transmission of response; completion of query; initiation of subquery.

The “Data set Class” is the model of database table with possible fragmentation. Its basic features are: table code; codes of storing nodes; volume; parent' table code. The class has methods for limitations verification: to check availability of at least one copy of each data fragment t and the total volume of the information kept on a node.

The general DDB object model is built as system of interactive object models of its typical components. The scheme of interaction of objects in the DDB model is represented in a fig. 1 as the UML diagram.

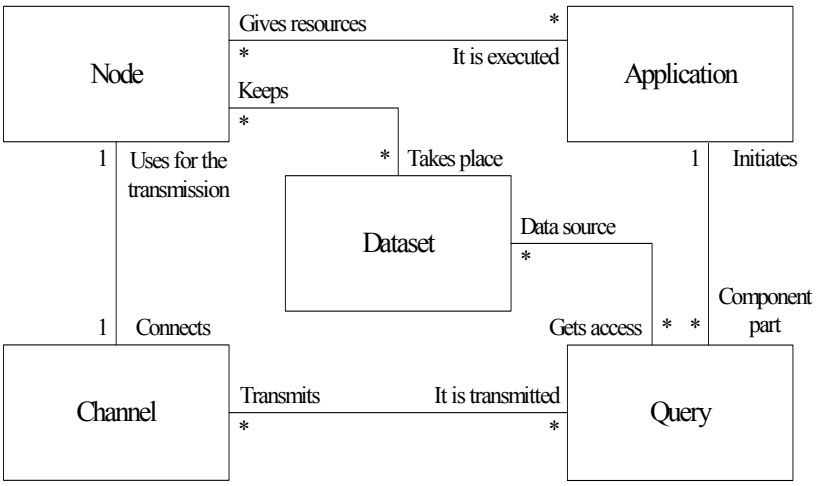

Figure 1: Scheme of interaction of objects in the DDB model

In the process of the DDB modeling the processes of queries been processed and updates been spread is simulated. Thus, the events time is calculated for each query or update: forming a queue for transmission; starting transmission; completing transmission; forming the queue for processing; starting processing; completing processing. These periods are fixed in the model table of events for further analytical processing. These values are used to calculate different parameters of DDB functioning, including estimation of the DDB efficiency criterion (1).

4 DDB optimization with the use of genetic algorithms and object model

To optimize data allocation on the nodes of the information system a new method has been developed. It is based on the use of genetic algorithms (GA) in combination with DDB object model [LT00, LT01]. The scheme of data fragments allocation on the DDB nodes is encoded as the set of chromosomes. GA population is the set of points of search space. Initial population is generated at random. Chromosomes are generated in the process of optimization carried out by GA operators (selection, crossover, mutation); these are the schemes of data sets allocation on the DDB nodes.

The schemes in focus are initial information for DDB object model. This information helps calculate the criterion of the DDB efficiency. These estimations, in their turn, are the values of GA fitness function for this way of solution. Scheme of interaction of the DDB object model with GA is shown in the fig. 2.

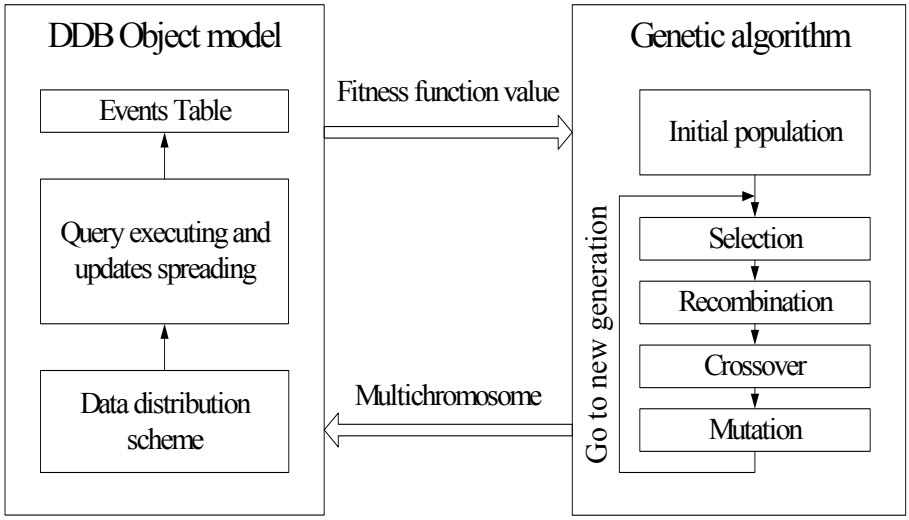

Figure 2: Scheme of interaction of the DDB object model with GA

A new modification of genetic algorithm has been developed to suit this task. It suggests using multichromosomal solution. The number of chromosomes necessary for encoding the scheme of data allocation on the DDB nodes will equal the number of nodes. Every chromosome is the vector of binary values, length of which comes up to the number of data fragments that are to be allocated (fig. 3).

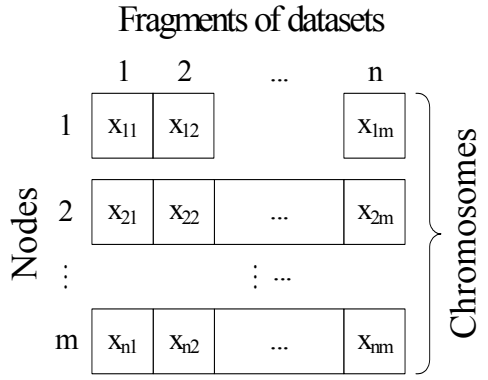

Figure 3: Structure of multichromosome

The genes of chromosomes (x_{ij}) take on value 1, if the copy of the corresponding data fragment is on a node and 0 if it is not. Thus, every chromosome encodes data fragments allocation on a particular node. The whole set of chromosomes fully encodes the scheme of data allocation in DDB.

For multichromosomal solution we have resorted to the operator of recombination of chromosomes sets having allowed us to accelerate the optimum decision search. This operator carries out the particular chromosome transfer from one solution to other. To make the operator of recombination function we have selected two individuals. Then each particular chromosome of an individual in a multichromosome with probability P_{rec} changes places with the proper chromosome of the other individual.

Taking into account the peculiar features of the solved task we have used the following standard GA operators: two point crossover, roulette method selection and mutation with the set probability.

Thus, the developed method of the DDB optimization, based on the use of modified GA and DDB object model, allows to determine the suboptimum schemes of data sets allocation on the nodes of the information system according to the criterion of minimum total average time required for queries processing and updates spreading.

5 Programming complex for DDB design and optimization

In accordance with the method of DDB optimization we have developed a programming complex. It allows to carry out DDB design and optimization aimed at improving its efficiency. The structure of programming complex is demonstrated in the fig. 4.

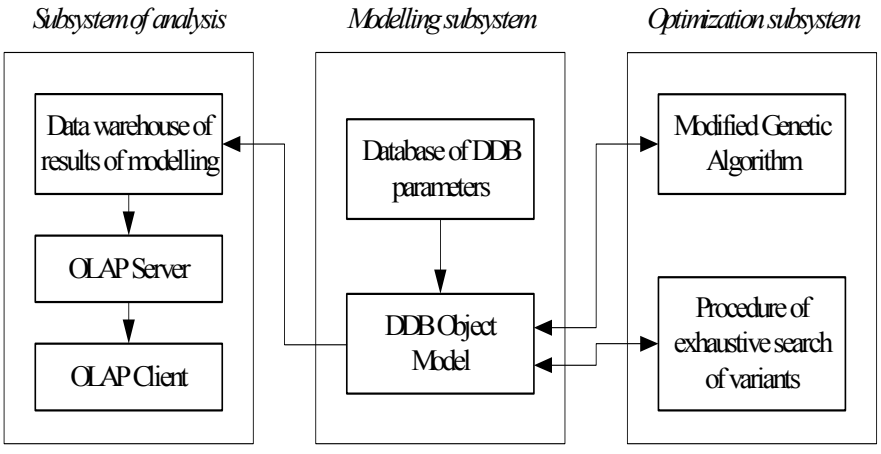

Figure 4: Structure of programming complex of the DDB modeling and optimization

The design subsystem is the program realization of the developed DDB object model. A model is realized with the use of visual programming system C++ Builder, together with storage database for initial data and modeling results developed on the basis of Microsoft Access DBMS.

The subsystem of modeling results analysis is developed on the basis of OLAP technology. Microsoft Analysis Services has been chosen to be the OLAP server, Microsoft Excel in its turn was used as the OLAP client. Microsoft Access tools have been used to create the storage database for the modeling results.

The subsystem of data allocation optimization is composed of two blocks. The first is in charge of program realization of optimization algorithm based on GA, that allows to find the suboptimum solutions, the second relates to program realization of scrupulous search procedure that allows to find the global optimum of the task.

6 Results of DDB optimization experiments

To carry out experimental researches of the developed method of the DDB optimization we used the computer information system of the "Kyev-Konty" company (Ukraine). This is a large blue chip confectionery manufacturer in Ukraine. The company consists of four factories: - 3 in Ukraine (Donetsk, Konstantynovka, Gorlovka) and one in Russia (Kursk). A company sales system is based on distribution. It has 5 subsidiaries (warehouses), four in Ukraine – in Donetsk, Kiev, L'vov, Nykolaev, and one more subsidiary in Russia is Voronezh, and also there are a few regional representative offices.

The information system of the "Kyev-Konty" company has the distributed architecture built on DDB, which consists of 10 nodes: central node (corporate server), each factory's node and a node in every subsidiary. The value of the DDB efficiency criterion (1) total average time required for queries processing and updates spreading in this system, was calculated by the DDB object model an came up to 111,77 sec.

To prove the efficiency of the developed method of the DDB optimization we have carried out a number of the computing experiments the results of which have been thoroughly analyzed. The modified GA and DDB object model allowed us to get a great number of suboptimum solutions (schemes of data sets allocation). These solutions were compared to the global optimum of the DDB efficiency criterion (1), calculated by scrupulous variants selection. As a result, we managed to calculate the minimum value of criterion T_{opt}, equal 79,1 sec. The time spent on the search for global optimum came up to approximately 17 days.

We also carried out the research defining the affect of population size and population number on the value of DDB efficiency criterion. Having analyzed the results we came to a conclusion that the proceeding values of population size $N_p = 60$ and number of generations $G = 20$ are the closest to the global optimum T_{opt}.

The way the value of the DDB effectiveness criterion T is affected by probability of mutation operators P_{mut}, recombination P_{rec} and crossover P_{cross} is demonstrated on the fig. 5-6. The analysis of the mentioned dependences proved that criterion T maximum approach to the optimum value T_{opt} takes place at the following probabilities: mutations $P_{mut} = 0,07$, recombination $P_{rec} = 0,5$, crossover $P_{cross} = 0,6$.

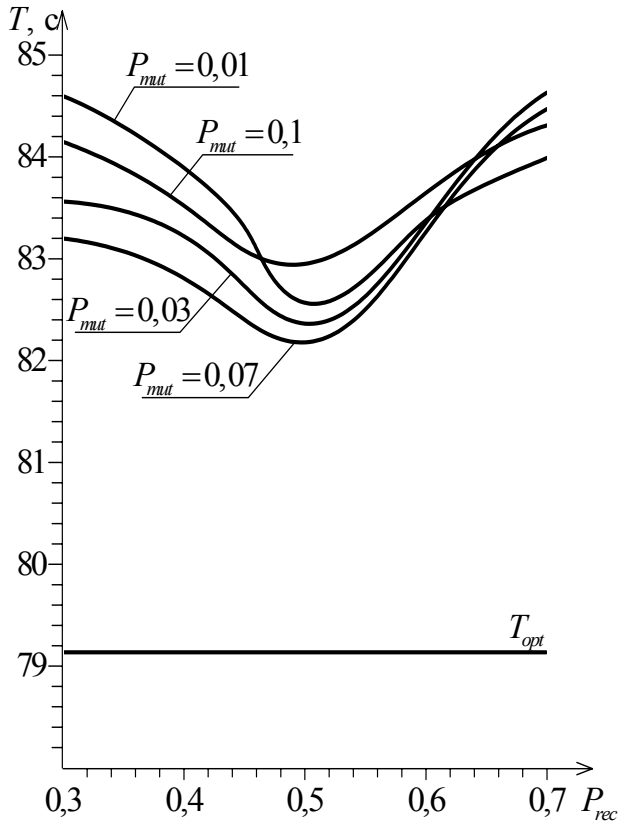

Figure 5: Dependence of DDB efficiency criterion T on probability of mutation operators application P_{mut} and recombination P_{rec}

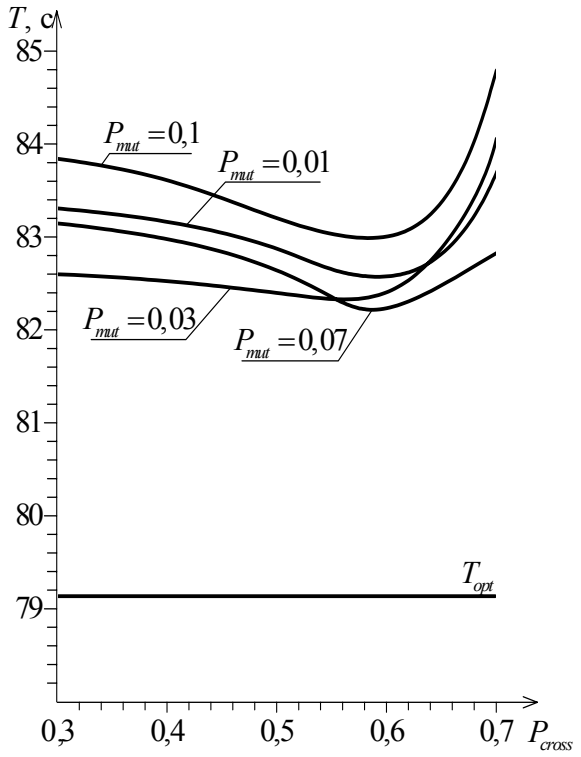

Figure 6: Dependence of DDB efficiency criterion T on probabilities of application of operators of mutation P_{mut} and crossover P_{cross}

To draw a conclusion, taking into account GA parameters defined by the experiment the best suboptimum value of DDB efficiency criterion is 82.19 sec. Absolute deviation of this value from a global optimum T_{opt} equals 3,09 sec., relative declination – 3,76%. As compared to the value of DDB efficiency criterion, calculated for the initial data allocation (111.77 sec.), its suboptimum value is less for 29.58 sec. or for 26.47%. The time required to define suboptimum decision using the developed method comes up to 2 minutes. (PC with the Intel Celeron 2.8 GHz processor.)

7 Conclusion

The paper under consideration suggests a solution for data set allocation optimization in the distributed database systems. The solution is based on object model of the distributed database and uses genetic algorithm. A sound solution allows to increase systems productivity without the need for additional expenses on hardware modification.

A programming complex for the modelling, analysis and the DDB optimization, which can be used both for development of new information systems with DDB and for the increase of the existing information systems productivity, is developed.

Bibliography

[Co00] Corcoran A.L., Hale J.: A genetic algorithm for fragment allocation in a distributed database system: Proceedings of the 1994 ACM symposium on Applied computing, Phoenix, ACM Press, 1994; pp. 247–250.

[Ga00] Galkin V.E.: The methods of optimal organization of distributed information system: Automation and modern technologies Journal, № 4, Moskow, Russia, 2004; pp. 13–17.

[LT00] Lazdyn S.V.; Telyatnikov A.O.: The distributed database optimization with using genetic algorithms: Journal of the Kherson State Technical University, volume 19, Kherson, Ukraine, 2004; pp. 236–239.

[LT01] Lazdyn S.V.; Telyatnikov A.O.: A new approach to the optimization of datasets allocation in computer information systems: Proceedings of the VI International scientific conferense “System analysis and information technologies”, Kiev, Ukraine, 2004; pp. 222–224.

[Te00] Telyatnikov A.O.: The obiect model of distributed database development: Journal of the Donetsk National Technical University, volume 74, Donetsk, Ukraine, 2004; pp. 192–200.

[Ts00] Tsegelik G.G.: Distributed database systems: L'vov, Ukraine,1990; 168 p.