The Optimization of Databases Distribution in a Distributed Computer Networks

Roman Krasniuk

Faculty of Applied Mathematics and Information Science, Ivan Franko National University of L'viv, UKRAINE, L'viv, Universytetska street 1, E-mail: krasniuk@ukr.net.

Abstract – ***In paper was considered mathematical model of optimization of databases distribution in a distributed computer networks. Made mathematical formulation of problems and formulate an efficient computational algorithm of solution of problems in the case of large dimension inputs. The computational algorithm is of type "greedy" algorithms to improve the procedure is performed, which provided a significant decrease in the average calculation errors to a level acceptable for practical calculations.***

Key words – optimization, distributed databases, computer networks, computer algorithms.

I. Introduction

Modern database management systems provide support for distributed databases. This is logically the only database of which can be placed in multiple network nodes. The nodes may be different types of computers that use different operating systems. Units can be geographically distributed but users receive the same access to all the information stored in the database. The main advantage of distributed systems is the reduction of expenditures for data transfer over the network, and the distribution of a large centralized database into multiple parts increases the efficiency of data processing.

The base aspects the optimization of network distributed systems was investigated in papers [1]-[3]. In particular, [1] conducted systematization of mathematical approaches and models for modeling of distributed databases. Mathematical models of multi synthesis of physical structures distributed databases were considered in [2]. Statement was formulated and proposed mathematical models offered to sell to computer networks of different topologies, criteria for minimum wage costs reduced access time to data and minimize network traffic. Concept of construction and choice of distributed databases of information retrieval systems dedicated to the work [3]. Questions algorithmic support distributed databases was considered in [4]. Consider the life cycle of designing a distributed database and describes its stages. The problem of distributed database design and assessed its complexity. The choice of genetic algorithms to solve the problem and proposed approach to take account of the interdependence of the design stage.

The authors of the direction of development and research of mathematical models of optimal distribution database is not yet sufficiently developed, despite the considerable number of works devoted to the analysis of various aspects of distributed databases. This situation is due to the fact that the formation of new architectural solutions raises the question of effective development management software such class of information systems. At the same time, the establishment of appropriate software impossible without the formation of the mathematical models and effective numerical algorithms. As a result, the construction and research of optimization of distribution databases is relevant and important issue, which is seen at work.

II. The mathematical formulation of the problem

Let n – number of computers of different capacities that make up a distributed computer network; m – number of databases that need to be placed in a computer network; x_i – number of databases that are planned for placement on the i – th computer network (the desired value); $c_i(x_i)$ – storage costs and maintenance x_j databases on the i – th computer network. Then a mathematical model of the problem will look like:

$$\sum_{i=1}^{n} c_i(x_i) \to min \tag{1}$$

under conditions

$$\sum_{i=1}^{n} x_i \to m. 0 \le x_i, i = 1, 2, ..., n. \tag{2}$$

In the case of resource constraints computer networks $(0 \le x_i \le l_i, i = 1, 2, ..., n)$ .Optimization problem (1) - (2) takes the form: to find a minimum of the objective function (1) restriction

$$\sum_{i=1}^{n} x_i \to m, 0 \le x_i \le l_i, i = 1, 2, ..., n. \tag{3}$$

III. Formation computational algorithm

Formulated model defines integer programming problem. Building a solution of this class applications is complicated by large dimension. Therefore, an effective approach is to develop approximate algorithms, including the use of "greedy" algorithms. These algorithms are intuitive heuristics in which every step of the decision that is most advantageous for this step, without what is going to find the next steps.

We describe the general scheme of the algorithm for integer programming problems (1), (2) and (1), (3), which uses the idea of "greedy" choice.

Let $X(x_i^*)$ – the set of acceptable plans of the original problem for which $x_i = x_i^*, I = \{1, 2, ..., n\}$ – a set of index variables, I_0 – set of indexes variables which procedure "greedy" choice assigned new values.

Algorithm.

Step 0. Let $X_0 = X, I_0 = \varnothing$, and choose the initial plan allowable $x^0 = \left(x_1^0, x_2^0, ..., x_n^0\right) \in X_0$. For each of n variables we find the limits: $z_i^L \le x_i \le z_i^R$ so that $x = \left(x_1^0, x_2^0, ..., x_n^0\right) \in X_0$ for all integer $x_i, z_i^L \le x_i \le z_i^R$. Limits can be set approximately irregularities.

Let formed set I_k and $X_k \in X$.

Step 1. If $I_k = I$, all the variables have new meaning. That is permissible constructed vector x, which is taken as approximate solution. Otherwise, proceed to step 2.

Step 2. For $j \in I \setminus I_k$ find

$$j_0 = \max_{j \in I \setminus I_k} \left\{ \min_{z_i^L \le x_i \le z_i^R} (c_1(z_1) + c_2(z_2) + \ldots + c_i(x_i) + \ldots + c_n(x_n)) \right\}$$

where $z_k = \begin{cases} x_k^0, k \in I \setminus I_k, k \ne j, \\ x_k^*, k \in I_k. \end{cases}$

Granted

$$x_{j_0}^* = \min_{z_{i_0}^L \le x_{i_0} \le z_{i_0}^R} (c_1(z_1) + c_2(z_2) + \ldots + c_{i_0}(x_{i_0}) + \ldots + c_n(x_n)),$$

$X_{k+1} = X_k\left(x_{j_0}^*\right), I_{k+1} = I_k \cup \{j_0\}$, and go to step 1.

Using an algorithm formulated, give solution to the problem (1), (3).

Solution begins with the initial plan acceptable $x = (0, 0, \ldots, 0)$.

Step 1: Calculate by j - th coordinate vector x: $x_j = l_j$, where $l_j = \max_i l_i$ and $c_j(l_j) \to \min$ provided that $\sum_{i=1}^{n} x_i \le m$.

Step 2. Transform the problem as follows: of the objective function and the left side constraints (3) exclude a member containing a change x_j, get that problem (4)

$$\begin{gathered} c_1(x_1)+c_2(x_2)+\ldots+c_{j-1}(x_{j-1})+c_{j+1}(x_{j+1})+\ldots+c_n(x_n)\to\min, \\ x_1+x_2+\ldots+x_{j-1}+\ldots+x_n\le m. \end{gathered} \tag{4}$$

As a result, change the number of variables in the initial problem is reduced by one. Go back to step 1.

Steps 1 and 2 are repeated for as long as all the variables vector x new values will not be broken or condition (3).

To evaluate the error of an approximate solution that got "greedy" algorithm was solved problems about 100 conditions that generated randomly. For exact solution used branch and bound method [5]. According to the results of empirical research mean error greedy solution is 15%.

To reduce the error of approximate solution procedure was proposed to improve it. In particular, further assume that variable vector x numbered in order of receipt values in the "greedy" algorithm.

Algorithm improvements solution.

As we take the original plan solution that is received by "greedy" algorithm. Holds $K = \{1, 2, \ldots, n\}, k = 1$.

Step 1: Choose not yet processed variable $x_k > 0$, with a minimum index value $k \in K$ and select $\delta_k > 0$, which is necessary to reduce the value x_k .

Step 2. Find the variable $x_q, q > k$ and value $\delta_q > 0$, which can reduce the value x_q without breaking the admissibility of the plan and at minimum objective function. For this δ_k and δ_q must satisfy the condition

$$c_q(\delta_q) < c_k(\delta_k) .$$

After step 2 whether or not the variable found x_q, pin $K = K \setminus \{k\}$, increase k by one and go back to step 1. The process stops when all the variables in the original plan will be reviewed $(K = \varnothing)$.

"Greedy" algorithm to improve the procedure was solved 150 tests whose terms were generated randomly. The results of computational experiments using the solution to improve procedures mean error of approximate solution dropped to 3.5%. That is, getting the results suggest, however, that the use of "greedy" algorithms is a promising tool for solving integer programming problems of large dimension.

Conclusion

The work was made setting the optimization problem of distribution of databases in a distributed computer network. Formulated efficient computational algorithm for constructing an effective solution of the problem in the case of large-scale input. The comparison obtained solution with the exact solution of the problem, which is obtained by the method of branches and borders. The high efficiency of the proposed algorithm computing solutions based on test objectives, inputs which generated randomly. Numerical experiments showed that the average error of the approximate solution does not exceed 4% of that is a good result for practical use.

Prospects of application of the work is the integration of mathematical models and computational algorithms to systems of resource management in distributed computer networks. Further research will continue to adapt computational algorithm to other models optimize resource allocation and changing computational algorithm to reduce the average error of the approximate solution.

References

[1] G. G. Tsegelik, "Distributed database systems", L'viv: Swet, 168 p., 1990.

[2] V. V. Beskorovainyi and O. S. Ulyanov, "Mathematical models of multi-criteria synthesis of physical structures of distributed databases", Eastern European Journal of advanced technologies., vol.4, pp.44-48, 2010.

[3] Y. S. Yakovlev, "On the concept of construction and the choice of distributed databases of information retrieval systems", Matematichni machines i system., vol.2, pp.35-53, 2013.

[4] E. I. Chumachenko and S. S. Zaharov, "Algorithmic support distributed databases", Artificial Intelligence., vol.1, pp.49-54, 2013.

[5] G. G. Tsehelyk, "Linear programming", L'viv:Swet, 206 p., 1995.