sciendo RIGA TECHNICAL UNIVERSITY 1862

Applied Computer Systems

ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
May 2019, vol. 24, no. 1, pp. 69–74
https://doi.org/10.2478/acss-2019-0009
https://content.sciendo.com

Placement of Data Array Replicas in a Distributed System With Unreliable Communication Channels

Vladimir Kulba[1], Sergey Somov[2*], Yuri Merkuryev[3]
[1]*Institute of Control Sciences of Russian Academy of Science, Moscow, Russia*
[2]*Institute of Control Sciences of Russian Academy of Science, Moscow, Russia*
[3]*Riga Technical University, Riga, Latvia*

***Abstract* – The paper proposes an approach to solving the problem of optimal placement of data array replicas in a distributed system. It presents a formal model of a distributed system with unreliable communication channels. A system is represented as a graph with coloured arcs. The formulation of the problem of finding the optimal placement of replicas is given. As a criterion for optimization, the minimum response time to a data request is considered. The task solution heuristic algorithm is also proposed in the paper.**

***Keywords* – Distributed systems, placement of data array replicas, unreliable communication channels.**

I. Introduction

Various requirements are imposed on the characteristics of distributed data processing systems (DDPS). The most important requirements are 1) processing of data requests within a reasonable time; 2) a high level of security of information used in the system; 3) high reliability of the system. An effective method to solve these requirements is to use data replication. Placing replicas of arrays of data closest to the nodes – sources of requests – can significantly reduce the system response time to such requests.

The tasks of finding the optimal location of replicas in distributed systems are computationally complex. This is especially true for large-scale systems. Therefore, to solve such problems, various methods are used to reduce their computational complexity [1]–[5]. Different criteria are used when formulating such tasks. The most common criteria are the minimum cost of system operation, the maximum probability of receiving a response to a user request and the minimum response time of the system to inquiries.

The article deals with the problem of finding the optimal distribution of replicas of several data arrays in the nodes of a distributed system operating on the basis of a computer network with unreliable communication channels. The minimum average response time of the system to queries against data arrays is used as the optimization criterion. The task uses restrictions on the maximum cost of the system and the limit on the maximum number of replicas located in one network node.

II. Distributed Data Processing System Parameters

The nodes of the distributed data processing system (user servers) run several application processes (tasks) of various types, which are initiated by users of the system. During the solution, each task generates information requests and/or queries to modify the data arrays. One information query requests data from a replica of a single data array. The data modification request is addressed to all nodes with replicas of the modified data array. If the solution of the problem requires data from several arrays, then it sequentially generates the corresponding number of queries.

Suppose that a distributed system operates on the basis of a computer network consisting of N nodes. The network topology is represented by a weighted graph $G = (X, \Gamma)$. The lengths of the graph arcs are specified. The system uses M $(m = \overline{1, M})$ data arrays of different types, differing by a set of record attributes. The data transmission network consists of $\Phi(\varphi = \overline{1, \Phi})$ communication channels. The capacity of each φ communication channel is equal to C_φ data units per unit time. Let $q^*_\varphi = 1 - \rho^*_\varphi$, $(\varphi = \overline{1, \Phi})$ is the probability of an error occurring in the φ-th channel. The success of the message delivery is confirmed by the transfer of ACK (ACKnowledgement) receipts. Let t_{ACK} be the waiting time for the sending node to receive the ACK receipt of message delivery to the receiving node. If during the time t_{ACK} the sending node does not receive an ACK receipt, the message is retransmitted.

In a distributed system, several user tasks are solved. Each task belongs to one of J possible types. The frequency of solving tasks is given by the matrix $F^* = \|f^*_{nj}\|$ $(n = \overline{1, N};\ j = \overline{1, J})$, where f^*_{nj} is the frequency of solving the task of the type *j* in the network node *n*. The matrix $E^* = \|e^*{}_{jm}\|$ determines the frequency $e^*{}_{jm}$ of the generation by the task of the type *j* of information requests to the array of the type *m*. The matrix $U^* = \|u^*{}_{jm}\|$ determines the frequency $u^*{}_{jm}$ of generation by the task of type j of the requests for modification of the data array m.

The system operates in a steady state, the nodes have endless queues for service, all requests are successfully processed. Suppose that the processing time of each request in any node of the system is the same and is equal to T_{pr}.

Information requests are sent along the shortest paths to the nearest nodes with replicas. The shortest paths between all pairs of vertices of a weighted graph G are determined using one of the known shortest-path search algorithms on weighted graphs [6]. The algorithm results in the matrix $SP = (sp_{nk})_{N\times N}$ of the shortest paths in the graph G, where the element sp_{nk} is equal to the length of the shortest path/route between nodes *n* and *k*.

[*] Corresponding author's e-mail: ssomov2016@ipu.ru

The distribution of replicas *M* data arrays across nodes of the system is described by the matrix $A = \|a_{nm}\|$, in which $a_{nm} \in \{0,1\}$, and $a_{nm} = 1$, if the network node n contains a replica of the array *m*. Each of *M* data arrays can have several replicas located on the nodes of the system. The number p_m of replicas of the array *m* in the system is $p_m = \sum_{n=1}^{N} a_{nm}$.

The elements of the three-dimensional matrix $B = \|b_{nkm}\|$ of "proximity" of the nodes of the network are calculated based on the *SP* and *A* matrices. The matrix *B* has the dimension *N* × *N* × *M*. The element of the matrix *B* is $b_{nkm} \in \{0,1\}$, and $b_{nkm} = 1$ if the replica of the data array *m* is located at node *k*, and the node *k* is the closest node for node *n*. In this case, information requests to the array *m*, initiated at the node *n*, are addressed to node *k* for processing.

The average length $\bar{l}_m^q$ of the information query to the data of the array *m* is known. The average length $\bar{l}_m^r$ of the response to this request is also known. The average length $\bar{\bar{l}}_m^q$ of the request for modification of the array *m* and the average length $\bar{\bar{l}}_m^r$ of the answer to this request are determined.

III. The Problem of Optimal Placement of Array Replicas in System Nodes

It is necessary to place the replicas of *M* data arrays in the nodes of the distributed system in a way that ensures the minimum value of the *F*(*A*) functional. The functional *F*(*A*) is the average time of processing queries generated by user tasks per unit time. The value of the functional consists of 3 components and is calculated as follows:

$$F(A) = \tilde{T} + T_{\text{pr}} + \tilde{T} \, , \tag{1}$$

where:

$\tilde{T}$ – average time to send a request from the user server to the system node with a replica of the required data array (replica server);

T_{pr} – average request processing time by the replica server;

$\tilde{T}$ – average time to send a response to a request from the replica server to the user server.

Thus, it is necessary to find the minimum of the functional:

$$\min F(A) = \min(\tilde{T} + T_{\text{pr}} + \tilde{T}). \tag{2}$$

In this task, the following restrictions apply:

Limit on the cost of OP_{cost} of the system (the cost of storing replicas and processing requests in the network nodes, and the cost of using the communication channels of the system):

$$OP_{\text{cost}} \leq COST_{\text{MAX}}. \tag{3}$$

Restriction RN_{MAX_n} to the maximum number of replicas of arrays that can be placed in separate nodes of the system:

$$\sum_{m=1}^{M} a_{nm} \leq RN_{\text{MAX}_n}, \qquad n = \overline{1,N} \tag{4}$$

IV. The Value of the Average Delay of Messages in the System

We define messages as information requests to the replicas of data, requests for modification of the replicas and answers to these questions (ACK receipts are not considered). Consider the message traffic between any two nodes *i* and *k*. In both nodes, user tasks can be solved. These tasks generate requests to the replicas. These two nodes can also store replicas of data arrays. Information requests from node *i* to node *k* are addressed if node *k* has a replica of the desired array, and it is the closest node for node *i*. Then the number y_{ik}^q of information requests and requests for data modification generated in node *i* is equal to:

$$y_{ik}^q = \sum_{j=1}^{J} f_{ij}^* \cdot \sum_{m=1}^{M} \left(e_{jm}^* \cdot b_{ikm} + a_{km} \cdot u_{jm}^*\right).$$

On the other hand, node *i* can also store replicas of data arrays and receive requests from node *k*. After their processing, the node *i* sends responses to the node *k* with an intensity equal to:

$$y_{ik}^r = \sum_{j=1}^{J} f_{kj}^* \cdot \sum_{m=1}^{M} \left(e_{jm}^* \cdot b_{kim} + a_{im} \cdot u_{jm}^*\right).$$

In total, the number of messages transmitted per unit of time from node *i* to node *k* equals

$$y_{ik} = y_{ik}^q + y_{ik}^r.$$

In the entire network, message traffic *Y* is generated per unit of time:

$$Y = \sum_{i=1}^{N} \sum_{k=1}^{N} \left(y_{ik}^q + y_{ik}^r\right).$$

The average length $1/\mu$ of messages transmitted over network communication channels is equal to:

$$1/\mu = Y^{-1} \left\{ \sum_{i=1}^{N} \sum_{k=1}^{N} \sum_{j=1}^{N} f_{ij}^* \left[\sum_{m=1}^{M} a_{km} \cdot d^1 + \sum_{m=1}^{M} a_{im} \cdot d^2 \right] \right\} \tag{5}$$

$$d^1 = \left(\bar{l}_m^q * {e^*}_{jm} \cdot sp_{ik} + \bar{\bar{l}}_m^q \cdot {u^*}_{jm}\right),$$

$$d^2 = \bar{\bar{l}}_m^q \cdot {e^*}_{jm} \cdot sp_{ki} + \bar{\bar{l}}_m^r \cdot {u^*}_{jm}.$$

To determine the average message delay T^* in the distributed system, we use the results published in [7], [8]. In these works, it is shown that in networks with average network connectivity (i.e., for networks in which a majority of nodes have more than one input and more than one output communication channel) each message originating from an internal network node has a random length with a distribution density: $p(b) = \mu e^{-\mu b}$ $(b \geq 0)$, where μ^{-1} is the average message length.

Let us assume that the message is processed according to the *FCFS* discipline [7], [9]. Then each φ th communication channel can be considered a queuing system (QS) of the form $M|M|1|\infty$. This QS has a Poisson input stream with an average value of γ_{ik} messages per second and an exponential distribution of service time with an average value of μ^{-1}/C_{φ}

(bits per second). Then, taking into account the assumptions made, the average message delay will be [7]:

$$T^{*} = \sum_{i=1}^{N}\sum_{k=1}^{N}\frac{\gamma_{ik}}{\gamma}Z_{ik}\,, \qquad (6)$$

where Z_{ik} is the average delay of the message transmitted from node *i* to node *k* along the shortest path sp_{ik}.

Communication channel φ is included in the path sp_{ik}, if messages transmitted along this path use this channel. Then we can assume that we are given a set of Φ matrices $X^{\varphi} = \left\|x_{ik}^{\varphi}\right\|, (\varphi = \overline{1,\Phi},\ i,k = \overline{1,N})$, in which the element $x_{ik}^{\varphi} = 1$ if the communication channel φ is a part of path sp_{ik}. Stream of messages passing through the channel φ has an intensity equal to:

$$\lambda_{\varphi}^{*} = \sum_{i=1}^{N}\sum_{k=1}^{N}\gamma_{ik}\,x_{ik}^{\varphi}\,.$$

Denote by T_{φ}^{*} the average time spent by a message waiting in a queue for service and transmitting it over the φ-th channel. Then the value Z_{ik} of the average delay of a message sent from node *i* to node *k* will be equal to the sum of times T_{φ}^{*} of its service in all channels included in the sp_{ik} route through which messages between these two nodes are transmitted:

$$Z_{ik} = \sum_{\varphi=1}^{\Phi} x_{ik}^{\varphi}\cdot T_{\varphi}^{*}\,.$$

Considering formula (6), the average message delay in the network is equal to:

$$T^{*} = \sum_{i=1}^{N}\sum_{k=1}^{N}\frac{\gamma_{ik}}{Y}\sum_{\varphi=1}^{\Phi} x_{ik}^{\varphi}\cdot T_{\varphi}^{*}\,.$$

After changing the order of summation, we have:

$$T^{*} = \sum_{\varphi=1}^{\Phi}\frac{T_{\varphi}^{*}}{Y}\sum_{i=1}^{N}\sum_{k=1}^{N}\gamma_{ik}\,x_{ik}^{\varphi}\,. \qquad (7)$$

Using formula (7), we get:

$$T^{*} = \sum_{\varphi=1}^{\Phi}\frac{\lambda_{\varphi}^{*}}{Y}T_{\varphi}^{*}. \qquad (8)$$

In formula (8), the average delay T^{*} of messages in the network is determined using the values of average delays T_{φ}^{*} of messages in communication channels.

Let us determine the value T_{φ}^{*}.

The communication channel φ can be considered a queuing system of the form $M|M|1|\infty$ with a Poisson input flow with intensity λ_{φ}^{*} and exponential distribution of service time with an average value μ^{-1}/C_{φ}. In [7], it is shown that in such a system the time of stay of the application in the system (time of delay of the message in the φ channel) is defined as follows:

$$T_{\varphi}^{*} = 1\ /\ \left(\mu C_{\varphi} - \lambda_{\varphi}^{*}\right). \qquad (9)$$

This formula is valid for networks with reliable communication channels. Then the average delay T^{*} of a message in a network with reliable communication channels will be equal to:

$$T^{*} = \sum_{\varphi=1}^{\Phi}\frac{\lambda_{\varphi}^{*}}{\gamma}T_{\varphi}^{*} = \frac{1}{\gamma}\sum_{\varphi=1}^{\Phi}\frac{\lambda_{\varphi}^{*}}{\mu C_{\varphi}-\lambda_{\varphi}^{*}}\,.$$

When transferring a unit of data in distributed systems with unreliable communication channels, an error may occur with the probability $q_{\varphi}^{*} = 1 - \rho_{\varphi}^{*}$ in the φ-th channel. The occurrence of an error leads to a re-transmission of the message. This increases the total service time of the message in the communication channel, and thereby reduces its throughput. The probability of $q_{\varphi} = 1 - \rho_{\varphi,}$ the occurrence of an error during transmission on the φ channel of one message having an average length of $l = \mu^{-1}$ units of data, according to formula (5), is equal to:

$$q_{\varphi} = 1 - \rho_{\varphi} = 1 - \left(1 - q_{\varphi}^{*}\right)^{l} = 1 - \left(\rho_{\varphi}^{*}\right)^{l}.$$

Then the average time t'_{φ} of the transmission of one message with the length $l = \mu^{-1}$ over an unreliable φ-th communication channel is equal to:

$$t'_{\varphi} = \rho_{\varphi}T_{\varphi}^{*} + \left(t_{ACK} + \rho_{\varphi}T_{\varphi}^{*}\right)q_{\varphi}\left(1 - q_{\varphi}\right),^{-1} \qquad (10)$$

where $q_{\varphi} = 1 - \left(\rho_{\varphi}^{*}\right)^{l}$.

Let C'_{φ} denote the actual throughput of the unreliable φ-th communication channel. The value of C'_{φ} depends on the reliability of the channel, its physical bandwidth C_{φ} and the value of time t_{ACK}.

The value of the average time t'_{φ} of sending a message over an unreliable communication channel is determined by formula (10). At the same time, it is equal to the ratio of the length of the message to the actual bandwidth of the communication channel, i.e.:

$$t'_{\varphi} = \frac{\mu^{-1}}{C'_{\varphi}}\,.$$

Then the actual bandwidth of the unreliable communication channel is equal to:

$$C'_{\varphi} = \frac{\mu^{-1}}{t'_{\varphi}}\ ,$$

where t'_{φ} is determined by formula (10).

The average time T'_{φ} of the message delay in the φ-th unreliable communication channel (by analogy with formula (9)) will be equal to:

$$T'_{\varphi} = 1\ /\ \left(\mu C'_{\varphi} - \lambda_{\varphi}^{*}\right).$$

Therefore, in a system with unreliable communication channels, the value $\tilde{T}$ of the average delay time of the message during its transmission over the network channels will be equal to:

$$\tilde{T} = \sum_{i=1}^{N}\sum_{k=1}^{N}\frac{\gamma_{ik}}{\gamma}\tilde{Z}_{ik} \text{ , where } \quad \tilde{Z}_{ik} = \sum_{\varphi=1}^{\Phi} x_{ik}^{\varphi} \cdot T'_{\varphi} \text{ .} \quad (11)$$

In this formula, $\tilde{Z}_{ik}$ is the value of the average message delay transmitted from node *i* to node *k* via unreliable communication channels of the network.

V. Cost of Distributed System Functioning

The cost of the functioning of a distributed system depends on several system parameters: the cost of data transmission via communication channels, number of replicas and their distribution across the system nodes, the cost of storing replicas in the system nodes, the intensity of requests to the system and the cost of processing them.

We will use the following notation:

- $dst(x_i, x_k)$ – length of the shortest path/route between the nodes x_i and x_k, equal to the value of the element sp_{ik};
- dtc – data transmission cost – the cost of transferring a unit of data along a unit length path;
- v^e, v^u– the average amount of data transmitted over the communication channels of the network when processing an information request and a request for data modification;
- $S\_cst_m(x_k)$ – the cost of storing one replica of the *m*-th array of data in the node x_k;
- $E\_cst_m(x_k)$, $U\_cst(x_k)$ – the cost of processing an information request and a request to modify the replica of the *m*-th array located in the node x_k;
- $W = \{w_1, \dots, w_k, \dots, w_N\}$ – the vector of "weights" of nodes of the graph. The weight w_k of the node *k* is equal to the average amount of data that the node *k* exchanges with other nodes of the system when processing requests are generated in this node per unit of time. Weight w_k is equal to:

$$w_k = \sum_{j=1}^{J} f_{kj}^{*} \left(\sum_{m=1}^{M} e^{*}{}_{jm}\, v^e + \sum_{m=1}^{M} u^{*}{}_{jm}\, v^u\right).$$

Using the matrix *A* and the set *X* of all network nodes, it is easy to obtain *M* sets X_m^r $(m = \overline{1, M})$ with the numbers of those nodes of *X* in which the replicas of *M* data arrays are located. The set X_m^r contains the p_m numbers of nodes of the set *X*, in which the p_m replicas of the data array *m* are located. These nodes process requests for replicas.

Denote by $d^m(x_n, X_m^r)$ the minimum distance from some node x_n of the set *X* to one of the nodes of the set X_m^r:

$$d^m(x_n, X_m^r) = \min_{x_j \in X_m^r} dst(x_n, x_j) \text{ ,} \quad m = \overline{1, M.}$$

For each set X_m^r we define a transmission number σ equal to the cost of data exchange between the nodes of the set X_m^r and nodes from the set *X* when processing queries at the nodes of the set X_m^r:

$$\sigma(X_m^r) = \sum_{n=1}^{I} d^m(x_n, X_m^r) \cdot dtc \cdot w_n \text{ .}$$

Then the costs $Cost_{dtr}$ of the system for data exchange between network nodes when processing requests per unit of time will be equal to:

$$Cost_{dtr} = \sum_{m=1}^{M} \sigma(X_m^r).$$

The cost of storing all the replicas of the data arrays located in the nodes of the system is equal to:

$$Cost_{St} = \sum_{n=1}^{N} \sum_{m=1}^{M} S_{cst_m}(x_k) \cdot a_{nm} \text{ .}$$

The cost of processing information requests and requests for data modification generated in the system is equal to:

$$Cost_{EU} = \sum_{n=1}^{N} \sum_{j=1}^{J} f_{nj}^{*} \cdot \sum_{m=1}^{M}\Big(e_{jm}^{*} \cdot \sum_{k=1}^{N} b_{nkm} \cdot E_{cst_m}(x_k) + u_{jm}^{*} \sum_{k=1}^{N} a_{km} \cdot U\_cst(x_k)\Big).$$

The total amount OP_cos of all costs for the operation of a distributed system is:

$$OP_{cos} = Cost_{dtr} + Cost_{St} + \ Cost_{EU}.$$

VI. Problem Solving Algorithm

The formulated task (2)–(3) has large computational complexity. Therefore, it is suggested to use the heuristic algorithm, which is described below. The flowchart of the algorithm is shown in Figure 1.

The detailed description of the algorithm:

- Step 1. We calculate the frequency fr_m of requests to each *m*-th array of data generated when solving user tasks in all *N* nodes of the system:

$$fr_m = \sum_{n=1}^{N} f_{nj}^{*} \left(v^{*}{}_{jm} + u^{*}{}_{jm}\right); \ \ m = \overline{1, M}.$$

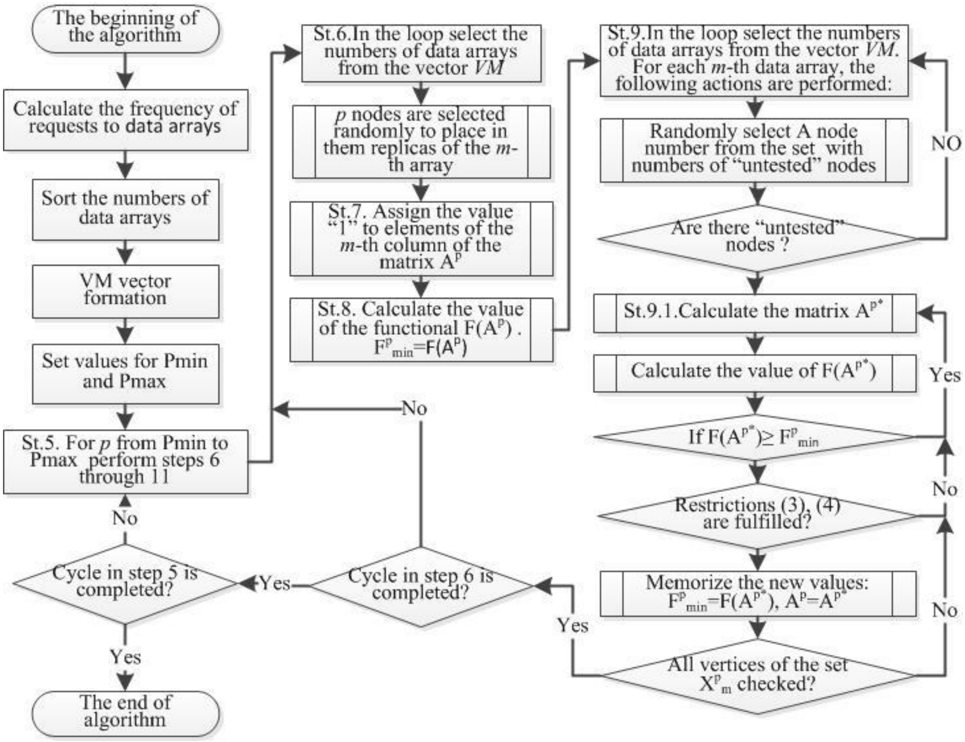
The beginning of the algorithm
Calculate the frequency of requests to data arrays
Sort the numbers of data arrays
VM vector formation
Set values for Pmin and Pmax
St.5. For p from Pmin to Pmax perform steps 6 through 11
No
Cycle in step 5 is completed?
Yes
The end of algorithm
St.6.In the loop select the numbers of data arrays from the vector VM
p nodes are selected randomly to place in them replicas of the m-th array
St.7. Assign the value "1" to elements of the m-th column of the matrix A^p
St.8. Calculate the value of the functional F(A^p). F^p_min=F(A^p)
No
Cycle in step 6 is completed?
Yes
St.9.In the loop select the numbers of data arrays from the vector VM. For each m-th data array, the following actions are performed:
Randomly select A node number from the set with numbers of "untested" nodes
Are there "untested" nodes ?
NO
St.9.1.Calculate the matrix A^p*
Calculate the value of F(A^p*)
Yes
If F(A^p*)≥ F^p_min
No
Restrictions (3), (4) are fulfilled?
Memorize the new values: F^p_min=F(A^p*), A^p=A^p*
No
Yes
All vertices of the set X^p_m checked?

Fig. 1. The flowchart of the problem-solving algorithm.

- Step 2. Sort the numbers of all M data arrays in descending order of their corresponding frequencies fr_m (m $= \overline{1,\mathrm{M}}$).
- Step 3. We form the vector $VM = \langle vm_k \rangle, k = \overline{1,\mathrm{M}}$. The element vm_1 of the vector contains the number of the array to which the largest number of queries is generated. Accordingly, the element vm_M contains the number of the data array to which the least number of queries is generated.
- Step 4. Set minimum $P_{\min}$ and maximum $P_{\max}$ number of network nodes, in which each of M replicas of data array may be placed.
- Step 5. In a loop by the number of p nodes with replicas, taking values from $P_{\min}$ to $P_{\max}$, we perform the steps of the algorithm from 6 to 11:
- Step 6. In the loop, one-by-one select the numbers vm_k $(k = \overline{1,\mathrm{M}})$ of data arrays from the vector *VM*.

For each of the next *m*-th data array, the following actions are performed:

  - Randomly, from the set of nodes X, p nodes are selected to place in them replicas of the *m*-th array.
  - The numbers of selected nodes form the set $X_p^m = \{x_{pi}^m \,|i = \overline{1,p}\}$. The element x_{pi}^m contains the node number i in which one of the replicas of the *m*-th data array is located. The numbers of these nodes are recorded into the set of X_m "*m*-tested" nodes. The remaining numbers of nodes of the set X that are not included in the set X_p^m are included into the set $\bar{X}_m$ of the numbers of "untested" nodes, i.e., $\bar{X}_m = \{X \backslash X_p^m\}$.

- Step 7. Assign the value "1" to those elements a_{nm}^p of the *m*-th column of the matrix A^p, whose number *n* of the row of the matrix is equal to the number x_{pi}^m of the network node with the replica.

The matrix $A^p = \|a_{nm}^p\|$ determines the distribution of p replicas of each of the M arrays over the network nodes. The element $A^p = \|a_{nm}^p\|$ of the matrix is equal to "1" if the replica of the m-th array is located in node n.

- Step 8. For the matrix $A^p = \|a_{nm}^p\|$ we perform the following operations:
  - Calculate the value of the functional $F(A^p)$ in accordance with formula (1).
  - Remember the resulting value: $F_{min}^p = F(A^p)$.
- Step 9. In the loop, one-by-one select the numbers vm_k $(k = \overline{1,\mathrm{M}})$of data arrays from the vector *VM*.
  For each of the next *m*-th data array, the following actions are performed:
  - From the set $\bar{X}_m$ with numbers of "untested" nodes we randomly select the network node number and store it in the variable x_i^{m*}.
  - If there are no more "untested" nodes, then return to Step 9.
  - Step 9.1. In the cycle, each vertex x_{pi}^m from the set $X_p^m = \{x_{pi}^m \,|i = \overline{1,p}\}$ is replaced by the vertex x_i^{m*}.
  - We get a new set X_p^{m*} in which one of the vertices is replaced by the vertex x_i^{m*}.
  - Create a copy of the matrix A^p as the matrix A^{p*}.
  - As in Step 7 assign the value "1" to the elements a_{nm}^{p*} of the *m*-th column of the matrix A^{p*} according to the values of the elements of the set X_p^{m*}.
  - Calculate the functional $F(A^{p*})$ for the matrix A^{p*}.
  - If $F(A^{p*}) \geq F_{\min}^p$ then return to Step 9.1.
  - Check the task constraints (3) and (4) for a new distribution A^{p*} of replicas.

- If one or both constraints are violated, we return to Step 9.1.
- Remember the new value of the functional and the new distribution of replicas:
- $F_{min}^{p} = F(A^{p*}), A^{p} = A^{p*}$
- If the loop on the vertices of the X_p^m is not completed, then we return to Step 9.

- Step 10. If not all the data arrays from the VM vector are viewed, go back to Step 6.
- Step 11. If the cycle by the number of p replicas is not completed ($p < P_{\max}$), then return to Step 5.
- Step 12. Search for replica distribution of M data arrays is completed.

The distribution of replicas A^p is found close to the optimal one over the nodes of the distributed system. This distribution provides the value of the average response time of the system to the request, equal to F_{min}^{p}.

VII. Conclusion

The article considers the problem of optimal placement of replicas of data arrays across nodes of a distributed system. The formulation of this task has several features that distinguish it from other tasks of this class. The formulation of the problem considers the possibility of errors during the transmission of requests and responses to them via communication channels. In the event of a failure in the channel, a message is sent again to the destination node, which leads to a decrease in system performance. The search for optimal replica placement is performed for several data arrays simultaneously. As a rule, in similar tasks the cost optimization criterion is used. In this paper, as a criterion for optimization, we consider the minimum average response time of a system to a request for data. An original heuristic algorithm for solving the formulated problem, implemented in C ++, is proposed. The use of this algorithm provides an improvement in the response time of the system by 5–8 % and reducing the cost of operating the system by 8–10 %.

References

[1] G. A. Chernyshev, "Obzor podhodov k organizacii fizicheskogo urovnja v SUBD" [Overview of approaches to the organization of the physical layer in the DBMS], *Proceedings of SPIIRAS*, St. Petersburg, vol. 24, pp. 222–275, 2013. https://doi.org/10.15622/sp.24.15

[2] A. Sleit, et al., "A Dynamic Object Fragmentation and Replication Algorithm in Distributed Database Systems," *American Journal of Applied Sciences*, vol. 4, no. 8, pp. 613–618, 2007. https://doi.org/10.3844/ajassp.2007.613.618

[3] T. Loukopoulos, I. Ahmad, and D. Papadias, "An Overview of Data Replication on the Internet," *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN.02)*, 6p. 2002.

[4] J. Sahoo, M. A. Salahuddin, and R. Glitho, "A Survey on Replica Server Placement Algorithms for Content Delivery Networks", IEEE Communications Surveys & Tutorials, 2016, p. 30. [Online]. Available https://arxiv.org/ftp/arxiv/papers/1611/1611.01729.pdf [Accessed: July 07, 2017].

[5] N. Mansouri, "Adaptive data replication strategy in cloud computing for performance improvement," *Frontiers of Computer Science*, vol. 10, no. 5, pp. 925–935, 2016. https://doi.org/10.1007/s11704-016-5182-6

[6] T. H. Cormen, et al. *Introduction to Algorithms, Third Edition*. The MIT Press, 2009.

[7] L. Kleinrock, *Queueing systems, Volume 2: Computer applications*. New York: Wiley, 1976.

[8] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Design*. McGraw-Hill, 1964.

Vladimir Kulba obtained the Doctoral degree in Technical Cybernetics and Information Theory in 1970, and Doctoral degree in Automated Information Processing and Control Systems in 1982 both from V. A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences.

Dr. Kulba is a Professor, Head of the Laboratory of Modular Information Management system of V. A. Trapeznikov Institute of Control Sciences of the RAS. His professional interests include theory and principles of constructing optimal modular data processing systems, methods for ensuring reliability in data processing, methods of redundancy of program modules and information arrays, development of theoretical bases of information security of systems, risk management, emergency management.

He has published more than 600 scientific publications, including more than 50 monographs and textbooks. For 25 years, under his leadership, an international conference "Problems of Security Managing of Complex Systems" has been held annually.

He is a member of the editorial boards of several scientific journals of the Institute. Vladimir Kulba is a full member of the Russian Academy of Natural Sciences and the International Academy of Informatization.

Contact information: V. A. Trapeznikov Institute of Control Sciences, Profsoyuznaya Str. 65, Moscow, 117997, Russia
E-mail: kulba@ipu.ru
Phone: +7 495 334-90-09
ORCID iD: https://orcid.org/0000-0002-5076-6536

Sergey Somov obtained the Doctoral degree in Automated Information Processing and Control Systems in 1984 from V. A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences.

His professional interests include data safety in distributed systems, data replication, backup and recovery of data in distributed systems. He has published more than 35 scientific publications, including 2 monographs and 1 scientific publication.

Contact information: V. A. Trapeznikov Institute of Control Sciences, Profsoyuznaya Str. 65, Moscow, 117997, Russia
E-mail: ssomov2016@ipu.ru
Phone: +7 495 334-90-09
ORCID iD: https://orcid.org/0000-0003-2966-9333

Yuri Merkuryev obtained the Dr. sc. ing. degree in System Identification in 1982, and Dr. habil. sc. ing. degree in Systems Simulation in 1997, both from Riga Technical University, Latvia.

Dr. Merkuryev is a Professor of the Department of Modelling and Simulation at Riga Technical University. His professional interests include modelling and simulation of complex systems, methodology of discrete-event simulation, supply chain simulation and management, as well as education in the areas of simulation and logistics management. Professor Merkuryev is a full member of the Latvian Academy of Sciences, fellow of the European Academy for Industrial Management, president of the Latvian Simulation Society, board member of the Federation of European Simulation Societies (EUROSIM), senior member of the Society for Modelling and Simulation International (SCS), and senior member of the Institute of Electrical and Electronics Engineers (IEEE). He is an associate editor of "Simulation: Transactions of The Society for Modelling and Simulation International", and editorial board member of "International Journal of Simulation and Process Modelling". He has authored more than 350 scientific publications, including 9 books and 6 textbooks.

Contact information: Riga Technical University, Kalku Str. 1, Riga, LV-1658, Latvia
E-mail: Jurijs.Merkurjevs@rtu.lv
ORCID iD: https://orcid.org/0000-0001-7178-5640