MULTIAGENT APPROACH TO COMPUTER MANAGEMENT IN A HETEROGENEOUS DISTRIBUTED COMPUTER ENVIRONMENT

Abstract. The reliability of modern computing control systems in a heterogeneous distributed computing environment, along with efficiency, survivability, security, control efficiency, is an important component of their quality. Increasingly, these systems fall into the category of “critical”, have an absolute impact on the activities of organizations and enterprises within which they operate. The loss of such systems, even for a short time, leads to serious problems related to loss of income, unforeseen costs, downtime of production and personnel, loss of time, and sometimes man-made disasters. As you know, the greatest impact on the reliability of control systems has the reliability and fault tolerance of a set of software and hardware. Therefore, solving problems related to improving the reliability of the software part of the systems is the most urgent task. Currently, significant results have been obtained in the field of evaluation and forecasting of reliability indicators of elements and typical software packages at the stage of their design; a large number of methods known to algorithms and programs are known; a number of normative documents on project reliability assessment have been developed. However, the task of real-time reliability assessment, when accurate and operational accounting of a number of factors is required, has not been sufficiently solved. To solve the problem of multi-agent approach to computing control in a heterogeneous distributed computing environment used methods of systems analysis, set theory - to develop models of task distribution, models of tasks and computing resources, general systems theory - to study and develop methods of task distribution, logic-theory theory, for modeling computational processes. The article considers a multi-agent approach to computing control in a heterogeneous distributed computing environment. The algorithm is based on the use of economic mechanisms to regulate the supply and demand of resources in the computing environment. The architecture of the multi-agent approach and the functions of the agents are described. Particular attention is paid to calculating the reliability of the task plan based on the logical-probabilistic method.

Keywords: multiagent control, agents, distributed computing, reliability, computational complexity.

Introduction

Currently, much attention in the field of Grid systems and cloud infrastructures is paid to the development of the fundamentals of distributed computing in solving large-scale scientific problems in various fields, as well as developing new methods and algorithms for managing computing in heterogeneous computing environment (HCE), complex hybrid structure. The hybrid node includes computing modules that support different parallel programming technologies and differ in their computational characteristics. Heterogeneity of computing environments, a wide range of tasks and the presence of hybrid nodes in their composition and the need to ensure the scalability of computing processes determine the variety of algorithms for managing distributed computing [1,3]. Decentralized multi-agent algorithms [4], as well as algorithms based on the use of elements of economic theory [5] show quite high efficiency.

High competition of user tasks for total resources of HCE leads to the need for comprehensive consideration of the characteristics of these resources in the process of their allocation in order to achieve the required quality of the task [6]. As a rule, in HCE as the main criteria of efficiency of management of distributed calculations such characteristics, as time and cost of performance of tasks of users, an indicator of balancing of loading of resources of environment and coefficient of their usefulness are applied. In addition to satisfying the above criteria for the quality of the task, one of the fundamental and practically important areas of research on the organization of distributed computing is also to ensure the reliability of the task in a heterogeneous computing environment. Improving the reliability of calculations allows you to better guarantee the fulfillment of other criteria for the quality of the task.

The article considers a multi-agent approach to the management of distributed computing, taking into account the reliability of HCE resources. It is assumed that the resources are provided by software agents, who are endowed with the necessary administrative rights and responsibilities and are grouped into virtual communities to perform user tasks. Task is a specification of the problem-solving process, which contains information about the required resources, running programs, input and output data, as well as other necessary information. If the task requires the execution of a number of interconnected modules, the computational process is carried out in accordance with the plan for solving the problem, which determines the order of execution of modules. The calculation of the reliability of the task plan is based on the logic-probability method [6], which is based on the transition from describing the conditions of reliability of a complex system using the functions of logic algebra to probability functions to determine indicators of this reliability.

Redundancy is one of the simplest and most effective methods of improving the reliability of computer systems [7]. In our case, the backup is the use
of the virtual community of agents of additional nodes, which in case of failure of the main nodes in the execution of the plan of the task can take over their functions. Additional nodes are a loaded reserve and are used in HCE to the same extent as the main nodes. The use of loaded backup is due to the fact that other types of backup, such as unloaded backup, hot backup or multi-task implementation [8], lead to significant overhead and are more suitable for real-time systems. The virtual community of agents uses the reservation of a specific node by the agent who selected that node. Because such redundancy may require too many nodes to achieve the required reliability, the article considers the problem of forming an additional set of nodes to achieve the reliability of the computational process, as close as possible to the specified reliability criterion, taking into account the number of nodes allocated by each agent.

Analysis of recent research and publications. Analysis of modern scientific and information sources in the field of research in the management of computations in a heterogeneous distributed computing environment Grid-systems of computational type. In researches Malashenko Yu.Ye., Konovalov M.G., Nazarova I.O., problems of task management in heterogeneous computer systems are considered [1]. The paper uses methods and tools for managing cloud computing systems [3] and services based on them [3, 5]. Viktortova V.S., Volik B.G., Stepanyants A.S., [6] focus their analysis on the reliability of the computing control system by a combination of computational models and logical-probabilistic method of studying the reliability of structurally complex systems [7] proposed by Cherechesov G.M. and Ryabinin I.A. According to the principle of construction and analysis, there are algorithms for synthesizing the architecture of a real-time computing system, taking into account the requirements for reliability according to Thomas H., Kostenko V.A., Zorina D.A. [8, 9]. Analytical control models [4] and reliability analysis of computing control complex [13] by a combination of computational models based on the economic mechanism of regulating their demand and supply based on security modeling technology of complex systems [14, 15] studied in Feoktistova AG, Bychkova I.V., Oparina G.A., Izmalkova S., Sonina K., Yudkevich M. Mozhayev A.S. The considered works of Yudkevich M. Mozhayev A.S. The considered works of Bychkova I.V., Oparina G.A., Izmalkova S., Sonina K., and uses a subset of parameters as output setting the desired values of the problem, and has an indicator of reliability - probability $p(t)$ execution of the plan for solving the problem (computational task) at time $t$ of receipt of the task. In this formulation, the task of planning calculations and resource allocation is NP (Non-deterministic Polynomial) - difficult [9].

In order to maintain the commonality of further considerations in the set F, two fictitious modules are introduced $f_1,f_2$, modeling the problem statement. Initial module $f_1$ has an empty set of input parameters, and uses a subset of parameters as output $Z_{in} \subseteq Z$, thus determining the condition of the task of the initial data of the problem. Target module $f_2$ receives a subset of parameters $Z_{out} \subseteq Z$, thus determining the condition for setting the desired values of the problem, and has an empty set of output parameters.

The set $S$ of plans for solving the problem can be represented by a bipartite oriented graph

$$G = \langle V, U \rangle$$

The set $V$ includes subsets $V_Z$ and $V_F$ vertices corresponding to the parameters with $Z$ and modules with $F$ included in the plans with $S$.

The set $U$ has two types of arcs between the vertices with $V_Z$ and $V_F$ : input arc connecting the vertex with $V_Z$ topped with $V_F$ and determining the relationship of the module with its input parameter, and the output arc connecting the vertex with $V_F$ topped with $V_Z$ and determining the relationship of the module with its output parameter.

We assume that the information-logical connections between the modules of the plan $s$ are described by a Boolean matrix $W$ of dimension $n_f \times n_f$, where $n_f$ - number of modules included in plan $s$. F is the set of software modules; $Z$ is the set of module parameters; $N$ is the set of HCE nodes; $A$ is the set of agents; $R_{an} \subseteq A \times N$ - the relationship between the elements of the sets $F, Z, N$ and $A$.

Relationships $R_{in}$ and $R_{out}$ type "many-to-many" set according to the input and output parameters of the modules and thus determine the information-logical connections between the modules. Attitude $R_{af}$ type "many-to-many" establishes a link between agents and modules that can be performed by agents. Attitude $R_{an}$ the one-to-many type identifies agent nodes. HCE computing clusters combine homogeneous nodes. Nodes of different clusters have different degrees of reliability. In the HCE computing management system, clusters are represented by agents.

The HCE receives a user request in a non-procedural form: "calculate parameter values from a subset $Z_{out} \subseteq Z$, by the values of the parameters from the subset $Z_{in} \subseteq Z$". In the general case, there may be many $S$ plans in the HCE model to solve this problem. Each of these plans determines which modules with $F$ and in what order should be executed.

You need to build sets $S$, choose a single plan from this set $s \in S$, identify agents with $A$ that will take part in its execution, and assign nodes with $N$ in which plan modules $s$ will be executed. The calculation process must meet the specified time or cost criteria for the effectiveness of the task, as well as strive for a certain indicator of reliability - probability $p(t)$ execution of the plan for solving the problem (computational task) at time $t$ of receipt of the task. In this formulation, the task of planning calculations and resource allocation is NP (Non-deterministic Polynomial) - difficult [9].
Matrix element \( w_{i,k} = 1 \) means that the module \( f_i \) depends on the input of the module \( f_k \).

**Computing control algorithm**

In this study, computational management is implemented by a multi-agent system.

The hierarchical structure of the system may include two or more levels of functioning of agents. At each level, there may be agents who play different roles and perform different functions. The roles of agents can be permanent and temporary, arising in discrete moments of time in connection with the need to organize collective interaction. Levels of hierarchy of agents differ in the amount of knowledge of agents - agents of the higher level of the hierarchy have more knowledge than agents of lower level of the hierarchy and, in addition, can ask agents of any lower level to obtain local knowledge of these agents.

At each level of the hierarchy, agents can come together in virtual communities, cooperate, and compete within those communities.

The multi-agent system includes computing and resource allocation planning agents. These agents are responsible for building a plan for the task and allocating the resources needed to perform it. The algorithm for constructing a solution plan is based on the use of the tender model of computational work [10] in the allocation of HCE resources and logical-probabilistic analysis of the reliability of the generated plan [6]. The combination of procedures for selecting a coordinator and conducting a tender significantly increases the efficiency of interaction between agents and the speed of decision-making. Here are the main stages of this algorithm:

- Actions of the calculation planning agent.
  - Construction of sets \( S \) of plans for solving the problem.
  - Formation of a virtual community of resource allocation agents who can participate in the implementation of the resulting set of plans.
  - SENDING the set \( S \) to the formed virtual community of agents.
  - Actions of resource allocation agents.
  - Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
  - Actions of resource allocation agents.
  - Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
  - Actions of resource allocation agents.
  - Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
  - Actions of resource allocation agents.

- Actions of the calculation planning agent.
- Construction of sets \( S \) of plans for solving the problem.
- Formation of a virtual community of resource allocation agents who can participate in the implementation of the resulting set of plans.
- Sending the set \( S \) to the formed virtual community of agents.
- Actions of resource allocation agents.
- Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
- Actions of resource allocation agents.
- Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
- Actions of resource allocation agents.
- Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
- Actions of resource allocation agents.

- Actions of the calculation planning agent.
- Construction of sets \( S \) of plans for solving the problem.
- Formation of a virtual community of resource allocation agents who can participate in the implementation of the resulting set of plans.
- Sending the set \( S \) to the formed virtual community of agents.
- Actions of resource allocation agents.
- Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
- Actions of resource allocation agents.
- Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
- Actions of resource allocation agents.
- Formation by agents of such information: ratings for selection of the agent-coordinator; intentions to implement all modules of the plan in the nodes of the agent.
- Actions of resource allocation agents.

In the implementation of this algorithm, presented in detail in [12], the main attention is paid to the economic mechanisms of regulation of supply and demand of resources.

As shown in [10], the use of economic mechanisms implemented within the algorithm and provide a system of penalties for agents for violating their nodes allowable limits of computational load, can improve the balancing of resource loads and increase their efficiency.

The following are new aspects of the algorithm related to the logical-probabilistic analysis of the reliability of the task plan.

**Logical-probabilistic model of reliability of the task plan**

Let:

- \( s \) be the solution plan chosen based on the results of the bidding.
- \( n_a \) - the number of virtual community agents involved in the plan \( s \),
- \( x \) is the set of Boolean variables \( x_{i,j,k} \), displaying execution events \( (x_{i,j,k} = 1) \) or non-compliance \( (x_{i,j,k} = 0) \) element \( f_i \) in the k-th node isolated by the agent \( a_j, i = 1, n_f, j = 1, n_a \).

Index \( k \) of the variable \( x_{i,j,k} \) determines the main accordingly \( (k = 1) \) and backup \( (k \geq 2) \) nodes agent \( a_j \), allocated for module execution \( f_i \).

Every agent \( a_j \) can highlight \( c_j \) backup nodes for plan \( s \) modules. It is assumed that all nodes of the same agent are homogeneous. The ability to run modules in the main or backup nodes causes different scenarios of the plan \( s \).

We introduce the following notation:

- \( y_i(x) \) - Boolean function, which determines the conditions of the module \( f_i \) in the process of calculating the plan \( s \);
- \( p_{i,j,k}(t) \) - the probability of execution of the module \( f_i \) in the k-th node of the agent \( a_j \).

The logic of the reliability of the plan \( s \) is described by formulas

\[
y_i(x) \equiv 1; \quad h_i(x), \text{ if } i = 2, (1)
\]

\[
y_i(x) = \left\{ \begin{array}{ll}
h_i(x), \text{ if } i = 2, x_{i,j,1} h_i(x), & \text{if } i > 2, (2) \\
0 &
\end{array} \right. 
\]

\[
h_f(x) = \bigwedge_{x_{i,j,k}=1} y_k(x), (3)
\]

where \( i \in 1, n_f, k = 1, n_f, j \in 1, n_a \).

First, in this scheme, the agent \( a_j \) and allocates to execute module \( f \) the main node.

Boolean function \( y_2(x) \), which determines the conditions of execution of the target module, is an indicator of the reliability of the plan \( s \).

After performing all substitutions according to formulas (1) - (3) function \( y_2(x) \) takes the form

\[
y_2(x) = \bigwedge_{i=3} x_{i,j,1}, (4)
\]
where $j_i \in \overline{1,n_a}$.

To calculate the reliability of the plan $s$, the transition $[13]$ is made using the appropriate rules from the function $y_2(x)$ to the probability function $P(t)$ of the following form:

$$P(t) = \prod_{i=3}^{n_f} p_{ijk1}(t).$$

Function $P(t)$ calculates the probability of execution of the only available scenario of the plan $s$. If $P(t) < p^* (t)$, then you need to convert the function $y_2(x)$ by improving the reliability of the elements of its structure.

Function $y_2(x)$, defined by formula (4), presented in disjunctive normal form without objections and, therefore, is monotonic.

This property ensures the absence of elements in its structure, for which the improvement of reliability deteriorates the reliability of the plan $s$ (study system) as a whole $[14]$.

Let, $J = \{ j : c_j = 0, j \in \overline{1,n_a} \}$ - a set of agent indexes that do not have backup nodes.

The process of transforming a function $y_2(x)$, corresponding to the plan $s$ without reserving nodes in the function $y_2'(x)$, corresponding to the plan $s$ with the reservation of nodes, includes the following steps. If

$$\sum_{j=1}^{n_a} c_j = 0,$$

that is, there is no possibility to reserve nodes, the completion of the transformation, otherwise - determine the index of the module with a minimum probability of its execution:

$$k = \arg\min_{i=3,n_f,j_i} \left( 1 - \prod_{i=1}^{e} (1 - p_{ijk1}) \right)$$

where $e = n_{i,j_i,i,j_i,i}j_i$ - the number of nodes allocated by the agent $a_{jk}$ and module $f_k$.

Reduction $c_{jk}$ per unit. Reserve an additional agent node $a_{jk}$ by removing the item $x_{k,jk,e}$ becomes a Boolean formula element, $x_{k,jk,e} \lor x_{k,jk,e+1}$, $e = n_{k,jk,e}, n_{k,jk,e} - the number of nodes allocated by the agent $a_{jk}$ module $f_k$.

Magnification $n_k$, $j_k$ per unit. Casting function $y_2'(x)$, obtained as a result of transformation, to the species:

$$n = \prod_{i=3}^{n_f} \sum_{j=1}^{n_a} n_{i,j_i},$$

$$y_2'(x) = \bigvee_{l=1}^{n} K_l,$$  \hspace{1cm} (6)

$$K_l = \bigwedge_{i=3}^{n_f} x_{i,j_i,e},$$

where $e = k_{j_i}$, $k_{j_i} \in \overline{1,n_{i,j_i}}$

Each elementary conjunction in formula (6) is one of the scenarios of possible implementation of the plan $s$. Elementary conjunctions of a function $y_2'(x)$ numbered from 1 to $n$ according to their rank in ascending order. In order to ensure the incompatibility of these scenarios, orthogonalization of the function is performed $y_2'(x)$ using the algorithm proposed in $[13]$.

Function $y_2''(x)$, orthogonal function $y_2'(x)$, takes the form

$$y_2''(x) = K_1 \lor K_2 \lor \ldots \lor K_{n_a} - 1K_n,$$

Simplify the function $y_2''(x)$ by removing identically equal zeros and absorbing conjunctions. Jump with the appropriate rules from the function $y_2''(x)$ to the probability function $P'(t)$ the following view:

$$P'(t) = \sum_{i=1}^{n} p_i'(t),$$

where $p_i'(t)$ - the probability of execution of the i-th scenario of the plan $s$.

Calculation of the reliability indicator of the plan $s$ using the function $P'(t)$.

If $P'(t) < p^* (t)$, then go to step 1. Otherwise - the completion of the transformation.

The above process of redundancy of nodes ensures the achievement of the reliability of the computational process, as close as possible to the specified criterion of reliability, taking into account the restrictions on the number allocated by each agent nodes.

These restrictions ensure the convergence of the node redundancy process.

**Conclusion**

Much literature has been devoted to the development of multi-agent distributed computing management systems, which to some extent touches on various aspects of their design, implementation and application. Analysis of the results of these studies shows that there are no finalized and established technologies for the organization of problematic HCE with multi-agent computing management. In this regard, we note a number of important features of the algorithm considered in the article.

In the course of work of algorithm efficiency of interaction of agents and speed of decision-making are provided. After the end of the auction, their participants reach a consistent steady state, which is to some extent analogous to the Nash equilibrium $[15]$, in game-theoretic models.

The algorithm allows to take into account different policies of HCE node administration, helps to improve load balancing and increase resource efficiency, provides a sufficient degree of fairness of the strategy of allocation of these resources based on specified time or cost criteria, and allows to achieve reliability of the task plan to a given criterion.

Further research in this area is related to the development of tools for flexible modernization of algorithms for the functioning of agents by connecting "external" libraries, including different scenarios of these algorithms.
Складовою їхньої якості. Все частіше ці системи потрапляють до категорії «критичних», тобто мають абсолютний вплив середовищі використані методи системного аналізу, теорії множин вирішення задачі мультиагентного підходу керування обчислюваннями в гетерогенному розподіленому обчислювальному реальному часу, коли необхідний точний та оперативний облік цілого ряду факторів, вирішено недостатньо. Низку нормативних документів щодо проектної оцінки надійності. Однак завдання оцінки надійності в масштабі комплексів на стадії їх проектування; відомо велике число методів, доведених до алгоритмів та програм; розроблено пов'язаних із підвищенням надійності функціонування програмної частини систем є найактуальнішим завданням. В даний систем управління надає надійність та відмовостійкість комплексу програмного і механізмів навігації та зв'язку, 2022, випуск 1(67) ISSN 2073-7394

Алгоритм базується на використанні економічних механізмів регулювання попиту і пропозиції ресурсів обчислювального середовища використання системного аналізу, теорії імовірнісного моделювання – теорії логічно-імовірнісного методу. Адаптація алгоритму до роботи в гетерогенному розподіленому обчислювальному середовищі відбувається через використання мультиагентного підходу до керування обчислювальними ресурсами, який включає імітаційні технології, теорію динаміки систем. Алгоритм базується на використанні економічних механізмів регулювання попиту і пропозиції ресурсів обчислювального середовищі. Описано архітектуру мультиагентного підходу і функції агентів. Особливу увагу приділено розрахунку надійності плану виконання завдання на основі логічно-імовірнісного методу.

Ключові слова: мультиагентне керування, агенти, розподілені обчисления, надійність, розрахунки надійності.