ISSN 8756-6990, Optoelectronics, Instrumentation and Data Processing, 2016, Vol. 52, No. 2, pp. 107–112. © Allerton Press, Inc., 2016.
Original Russian Text © I.V. Bychkov, G.A. Oparin, A.G. Feoktistov, I.A. Sidorov, V.G. Bogdanova, S.A. Gorsky, 2016, published in Avtometriya, 2016, Vol. 52, No. 2, pp. 3–9.

ANALYSIS AND SYNTHESIS OF SIGNALS AND IMAGES

Multiagent Control of Computational Systems on the Basis of Meta-Monitoring and Imitational Simulation

I. V. Bychkov, G. A. Oparin, A. G. Feoktistov, I. A. Sidorov, V. G. Bogdanov, and S. A. Gorsky

Matrosov Institute for System Dynamics and Control Theory, Siberian Branch, Russian Academy of Sciences, ul. Lermontova 134, Irkutsk, 664033 Russia
E-mail: oparin@icc.ru

Received September 3, 2015

Abstract—This paper describes the control of computations in a distributed computing environment (DCE) on the basis of its meta-monitoring and simulation modeling. Computations are controlled by a multiagent system with a given organizational structure. Resource allocation is carried out by agents with the use of economic mechanisms for controlling their supply and demand. Controlling actions for agents are formed on the basis of the simulation modeling of functional processes of the DCE. Data about the DCE resources and processes are collected and emergency situations in the DCE nodes are detected and prevented by the meta-monitoring system of this environment. The research results are the techniques for selecting control actions and the methods for intellectual processing and effective storage of data.

Keywords: distributed computing, multiagent control, monitoring.

DOI: 10.3103/S8756699016020011

INTRODUCTION

The analysis of the development of trends in high-performance computing both in Russia and abroad [1] suggests that controlling parallel and distributed computer systems is currently one of the most important fundamental problems. The role of a computing system in this paper is played by a distributed computing environment (DCE) with heterogeneous nodes known as computing clusters (CCs) with a complex hybrid structure. A hybrid cluster includes computing modules (hardware components) supporting various parallel programming techniques and differing in their computational performance. A distributed computing environment has a number of properties that significantly complicate the unification of computation scheduling and resource allocation and the estimation of effectiveness and reliability of the environment operation [2]. It is possible to identify a number of successful studies in the field of distributed computing control [3–5], including the effective solution of problems using graphical processor units [6]. However, the use of multiagent technologies, monitoring means, and DCE modeling makes it possible in some cases to obtain more effective results of computing control and to improve the DCE reliability.

The aim of this study is to develop a control system for computations in the DCE, which would implement new original multiagent methods and means for computation scheduling and resource allocation on the basis of meta-monitoring and simulation modeling of the DCE; as the results of computational experiments show, all of the above-mentioned ensures the high performance of computing control and analysis of the DCE reliability.

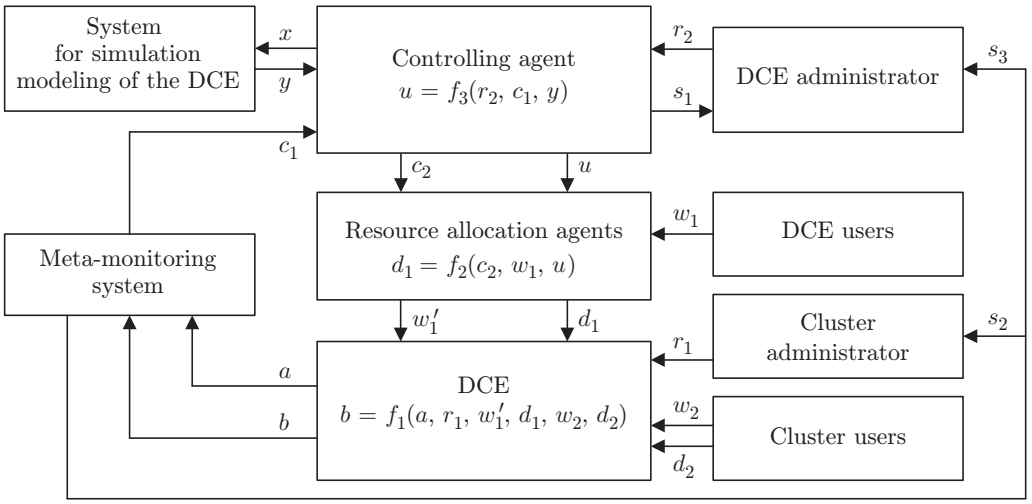

Fig. 1.

COMPUTING CONTROL SYSTEM

Currently, there are various agent-based methods and means of computing control [7, 8] used in practice, including those based on the economic principles of computing resource allocation [9–11]. However, the features of the DCE under consideration [2] do not allow applying them with sufficient effectiveness.

In the framework of the proposed research, the computing control is carried out with a multiagent system (MAS) with a given organizational structure. The actions of agents are coordinated with the use of group behavior rules. Agents operate in accordance with assigned roles. Each role has its own behavior rules in the virtual community (VC) of agents. A multiagent system includes the agents of meta-monitoring and resource allocation and a control agent. Resource allocation agents may be combined in a VC. In different VCs arising in MAS's, agents coordinate their actions by cooperation or competition.

The aim of an MAS is to obtain the allocation of job flows going into the DCE, which keeps the quality indicators of the DCE operation within the limits set by the DCE administrator. A job is specification of the problem-solving process that includes information on the required computing resources, executable application programs, input/output data, and other necessary information.

A block diagram of the computing control system is shown in Fig. 1. In the diagram, the DCE serves as a control object for which the job flows w_1 and w_2 of the DCE and VC users are external disturbances, the allocation results d_1 and d_2 of the flows w_1 and w_2 in the VC are the control actions of the MAS and of the VC users, respectively, and the vector r_1 of the parameters of the administrative policies of the VC is a corrective action. The resource allocation agents intercept the jobs of the flow w_1 for better configuration of the DCE requirements contained in the jobs, while the flow w_1 is modified into the flow w_1'. The allocation d_1 of the flow w_1' is produced by the agents based on economic mechanisms [12]. The allocation d_2 of the flow w_2 is determined by the VC users.

Information about the computing characteristics of the DCE nodes and the current performance indicators in these nodes are collected by the meta-monitoring system agents [13] with the help of control and measuring means in the form of the data structures a and b. There is abstract relation between the components of the structure b, on the one hand, and the elements of the structure a, the corrective action r_1, the job flows w_1' and w_2, and the allocations d_1 and d_2, on the other hand: $b = f_1(a, r_1, w_1', d_1, w_2, d_2)$. For various components of the structure b, this relation is represented by a functional, statistical, ambiguous, or some other type of mapping. The relations f_2 and f_3 have the same nature as the relation f_1.

The collected information in the form of the vector c_1 of aggregated indicators of performance of the control object are taken to the controlling agent at its request sent with a certain period of discreteness T_q. The value of T_q is chosen so as not to overload the DCE communication network and, at the same time, to accurately capture the moments when the performance indicators of the control object approach their limit values. Part of the information submitted by the vector c_1 and relevant to the resource allocation agents is immediately taken to these agents in the form of the vector c_2.

The vector r_2 of parameters of the administrative policies of the DCE is a corrective action for the controlling agent. Based on the information provided by the vectors c_1 and r_2, the controlling agent predicts the dynamics of the performance indicators of the control object at specified times for a certain time interval with the use of a simulation modeling system with a period of discreteness $T_s > T_q$. The modeling results are used to create the vector of control actions u on the algorithm of operation of the resource allocation agents of the VC by means of parametric adjustment of the algorithms. After the vector u is formed, it is transferred to each VC.

SIMULATION MODELING OF THE DCE

Let x and y be the vectors of input variables and observable variables of the simulation model of the DCE. The observed variables are the performance indicators of the DCE. The elements of the vectors x_i, $i = \overline{1, n_x}$, and y_j, $j = \overline{1, n_y}$, have the respective domains X_i and Y_j. The effects of input variables on the observed variables are studied by means of factor analysis in advance: at the time of construction and testing of the simulation model of the DCE. It is also assumed that each jth element of the vector y corresponds to the estimation criterion $\hat{y}_j$ of the value quality of this element (tendency to a minimum or maximum value) and its limit values $y_j^{\min}$, $y_j^{\max} \in Y_j$. A number of elements of the vector x serve as variable quantities, form a subset X^*, and are identified with the elements of the vector u: $u_q \equiv x_i$, $q = \overline{1, n_u}$, $i \in \overline{1, n_x}$, and $1 \le n_u < n_x$. The initial values of the variable quantities are basic values that correspond to the performance parameters of the DCE accepted by default. Subsequent values are selected from their domains in view of the effect of x_i to y_j, $i = \overline{1, n_x}$, $j = \overline{1, n_y}$. The values of the unvariable quantities which are the elements of the vector x are defined from the numerical information provided by the vectors r_2 and c_1.

In the process of modeling, a set of V variants of values of the observed variables is formed: the variable $y_{jk} \in Y_j$ is an element of kth variant $v_k \in V$ for the variable y_j, $j = \overline{1, n_y}$, $k = \overline{1, n_v}$. The formation of the subset $V^* \subseteq V$ of variants for the observed values on the basis of the set V in order to further determine the elements of the vector u is multi-criterion formation. If the observed variables are ordered by importance, the variants for V^* are selected on the basis of the lexicographical method; otherwise, on the basis of the majority method. The lexicographic and majority methods use the following selection rules [14]:

$$V^* = \{v_k \in V : (\forall v_l \in V \exists p \in \overline{1, n_y - 1} :$$

$$(\hat{y}_{1k} = \hat{y}_{1l}) \wedge \ldots \wedge (\hat{y}_{pk} = \hat{y}_{pl}) \wedge (\hat{y}_{(p+1)k} > \hat{y}_{(p+1)l}))\}. \quad (1)$$

Here $y_j^{\min} \le y_{jk} \le y_j^{\max}$, $j = \overline{1, n_y}$; $k \in \overline{1, n_v}$; $l \in \overline{1, n_v}$; $k \ne l$,

$$V^* = \left\{v_k \in V : \left(\neg \exists v_l \in V : \sum_{j=1}^{n_y} \operatorname{sign}(\hat{y}_{jl} - \hat{y}_{jk}) > 0\right)\right\}, \quad (2)$$

where $\operatorname{sign}(0) = 0$; $y_j^{\min} \le y_{jk} \le y_j^{\max}$; $k \in \overline{1, n_v}$; $l \in \overline{1, n_v}$; $k \ne l$.

To estimate the values of y_{jk}, $k = \overline{1, n_v}$, of the jth variable, their set is divided into subsets that do not intersect pairwise and are ordered by ascending or descending order. Accordingly, each subset receives its index used as an estimate of the observed variables belonging to a given subset.

Let V^* include the only variant v_k which is consistent with the kth set of variable quantities of the vector x. Selecting x_{ik} (with $x_i \in X^*$) from them, we obtain the values of the elements of the vector u. If the number of variants in V^* is larger than one, then a single v_k is selected randomly. With $V^* = \oslash$, the controlling agent generates a signal s_1 that requires new corrective actions from the DCE administrator.

META-MONITORING SYSTEM FOR THE DCE

Just like any other technical systems, the DCE nodes and their components are known to fail. With a large number of the DCE nodes, such emergency situations terminate the operation of parallel jobs and require restarting them, usually with the help of checkpoints. Restarts are accompanied with additional overhead costs and reduce the DCE performance [15].

A meta-monitoring system for the the DCE is designed to detect and prevent emergency situations in the nodes. With a certain period of discreteness T_i, the meta-monitoring agents measure the value of h_i of the

Control system	Performance indicators of the computing control system					
	n_{avg}, units	t_{avg}, s	k_{avg}, %	σ	n_{rest}, units	n_{err}, units
GridWay	520.362	701.750	0.690	0.005	123	31
MAS	98.071	286.184	0.746	0.003	57	1

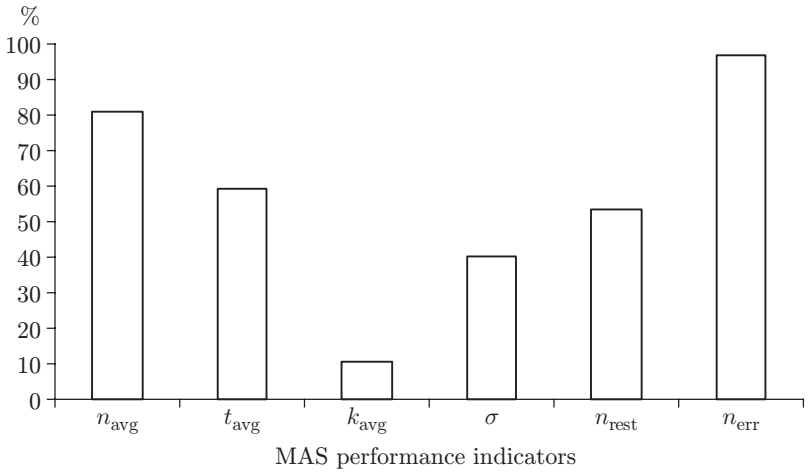

Fig. 2.

DCE node using local monitoring systems m_j (e.g., Ganglia and Nagios [16]) and controlling and measuring means e_k (for example, Sensors, APC PowerChute, IPMI, and SMART [17]), $i = \overline{1, n_h}$, $j = \overline{1, n_m}$, $k = \overline{1, n_e}$. Each ith characteristic has its limit values $h_i^{\min}$ and $h_i^{\max}$: $h_i^{\min} \le h_i \le h_i^{\max}$. The entire set of the node characteristics is divided into three subsets:

(1) the computational load volumes of the node components (processors, cores, memory, network components, data storage systems, and other structural elements);

(2) the physical state of the node components (the temperature of CPU and motherboard and the performance of uninterruptible power supply systems, hard drives, and other structural elements);

(3) the job implementation process (job priority and status, CPU time, amount of memory, the number of accesses to the hard drive and network elements, and other information).

Due to the large amount of information on the characteristics of the DCE nodes, the meta-monitoring system has an original round robin database [18], which demonstrated high effectiveness of the processes of accumulation, unification, and aggregation of information in comparison with similar known bases, such as MRTG [19] or RRDtool 20]. On the basis of the information obtained, the characteristics of the DCE nodes are expertly analyzed in the meta-monitoring system. If the value of h_i approaches its limit values $h_i^{\min}$ or $h_i^{\max}$, the meta-monitoring system generates signals s_2 and s_3, requiring new corrective actions from the administrator of the cluster included in the node and from the new administrator of the DCE. Analyzing how the jobs are accomplished in the nodes makes it possible to also detect the ineffective operation of user applications and to account for these results by the resource allocation agents. For example, in the Maker software package [21], the operation of saving information in a network directory dominates over computational operations, which is due to the low performance of systems for storing information in the nodes. Starting jobs in the nodes with connected local storage improves the effectiveness of the Maker software package by more than 30%. An expert subsystem is implemented in the CLIPS environment [22]. The periods of discreteness of measurements of characteristics and their expert analysis provide an operational response of the meta-monitoring system to emergency situations with the DCE hardware, thereby increasing the reliability and effectiveness of the DCE performance.

RESULTS OF COMPUTATIONAL EXPERIMENTS

In order to study the above-discussed methods and tools more thoroughly, we perform simulation modeling of the DCE performance using the GPSS World system [23]. The simulated system consists of 10 clusters (with the number of cores from 6000 to 14 000 units) and 300 users. The total number of cores is 100 000 units. The clusters include hybrid nodes supporting various parallel programming techniques. In the case of simulating the time it takes to perform a job, the count acceleration coefficient is used, whose values ranged from 1 to 1.5 for different clusters, depending on their computing characteristics. The simulated time of the system operation is 30 days. During this period, 12 990 flows are processed, including from 1000 to 10 000 processes for parallel programs or multivariant computations. At the computing control systems used in the DCE are the GridWay meta-scheduler [24] and the MAS presented in this paper. The job queue discipline is FCFS (First Come, First Served) with priorities. The main observed variables of the simulation model are the following average values: n_{avg} is the number of jobs in the cluster queue, t_{avg} is the stay time of the job in the cluster queue, k_{avg} is the effectiveness of using the cluster nodes, n_{rest} is the number of the program restarts, n_{err} is the number of failed jobs, and σ is the standard deviation factor of the useful cluster nodes. The simulation results given in the table show that the use of the MAS can significantly improve all the selected indicators of the DCE operation in comparison with the GridWay meta-scheduler (Fig. 2).

CONCLUSION

This paper describes the problem of computing control in the DCE on the basis of its meta-monitoring and simulation modeling. The original multiagent computing control system is created. The technique of multi-criteria selection of controlling actions of the MAS is proposed. The effectiveness and reliability of the DCE performance are improved by the methods of decentralized intelligent processing and distributed data storage. The simulated computing experiment is carried out. It is shown that the created MAS is better than the GridWay meta-scheduler widely used in practice by a set of important indicators of the computing control effectiveness.

This work was supported by the Russian Foundation for Basic Research (Grants No. 15-29-07955-ofi_m and No. 16-07-00931-a).

REFERENCES

1. A. V. Shamakina, "Overview of Distributed Computing Technologies," Vestn. Yuzhn.-Ural. Gos. Univ. Ser. Vychislitel'naya Matematika i Informatika **3** (3), 51–85 (2014).
2. V. G. Bogdanova, I. V. Bychkov, A. S. Korsukov, et al., "Multiagent Approach to Controlling Distributed Computing in a Cluster Grid System," J. Comput. Syst. Sci. Intern. **53** (5), 713–722 (2014).
3. V. N. Kovalenko, E. I. Kovalenko, D. A. Koryagin, et al., "The Fundamental Principles of the Advanced Planning Method for Computational Grids," Vestn. Sam. Gos. Univ. Estestvenno-Nauchnaya Ser., No. 4, 238–264 (2006).
4. M. G. Konovalov, Yu. E. Malashenko, and I. A. Nazarova, "Job Control in Heterogeneous Computing Systems," J. Comput. Syst. Sci. Intern. **50** (2), 220–237 (2011).
5. V. V. Toporkov, "Job Control in Distributed Environments with Non-Dedicated Resources," J. Comput. Syst. Sci. Intern. **50** (3), 413–428 (2011).
6. A. A. Yakimenko , K. V. Gunbin, and M. S. Khairetdinov, "Search for Overrepresented Characteristics of Genes: Implementation of Permutation Tests Using GPUs," Avtometriya **50** (1), 123–129 (2014) [Optoelectron., Instrum. Data Process. **50** (1), 102–107 (2014)].
7. E. H. Durfee, "Distributed Problem Solving and Planning," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, Ed. by G. Weiss (MIT Press, Cambridge, 1999).
8. T. Altameem and M. Amoon, "An Agent-Based Approach for Dynamic Adjustment of Scheduled Jobs in Computational Grids," J. Comput. Syst. Sci. Intern. **49** (5), 765–772 (2010).
9. A. Mutz, R. Wolski, and J. Brevik, "Eliciting Honest Value Information in a Batch-Queue Environment," in *Proc. of the 8th IEEE/ACM Intern. Conf. on Grid Computing* (IEEE, 2007), pp. 291–297.
10. *Market-Oriented Grid and Utility Computing*, Ed. by R. Buyya and K. Bubendorfer (Wiley & Sons, Hoboken, 2010).
11. V. V. Toporkov and D. M. Yemelyanov, "Economic Model of Scheduling and Fair Resource Sharing in Distributed Computations," Programming and Computer Software **40** (1), 35–42 (2014).

12. I. V. Bychkov, G. A. Oparin, A. Feoktistov, et al., "Multiagent Algorithm for Computing Resource Allocation on the Basis of the Economic Mechanism Regulation of Supply and Demand," Vestn. Komp'yuternikh i Informatsionnykh Tekhnologii, No. 1, 39–45 (2014).
13. G. A. Oparin, A. P. Novopashin, and I. A. Sidorov, et al., "Meta-monitoring System for Distributed Computing Environments," // Programmnye Produkty i Sistemy, No. 2, 45–48 (2014).
14. L. A. Sholomov, *Logical Methods for Studying Discrete Choice Models* (Nauka, Moscow, 1989) [in Russian].
15. V. B. Betelin, A. G. Kouchnirenko, and G. O. Raiko, "Problems of Productivity Growth in Domestic Supercomputers Until 2020," Informatsionnye Tekhnologii i Vyschislitel'nye Sistemy, No. 3, 15–18 (2010).
16. S. Zanikolas and R. Sakellariou, "A Taxonomy of Grid Monitoring Services," Future Generat. Comput. Syst. **21** (1), 163–188 (2005).
17. K. Charoenpornwattana, *A Scalable Unified Fault Tolerance for High Performance Computing Environments* (Louisiana Tech University, Ruston, USA, 2008).
18. I. A. Sidorov, A. P. Novopashin, G. A. Oparin, et al., "Methods and Means of Distributed Computing Environments," Vestn. SUSU. Ser. Computational Mathematics and Informatics **3** (2), 30–42 (2014).
19. MRTG — The Multi Router Traffic Grapher. http://oss.oetiker.ch/mrtg.
20. *RRDTool.* http://www.rrdtool.org.
21. *MAKER — Genome Annotation Pipeline.* http://gmod.org/wiki/MAKER.
22. A. P. Chastikov, D. L. Belov, and T. A. Gavrilova, *Development of Expert Systems. CLIPS Environment* (BKhV-Peterburg, Saint-Petersburg, 2003) [in Russian].
23. V. D. Boev, *System Simulation. GPSS World Tools* (BKhV-Peterburg, Saint-Petersburg, 2004) [in Russian].
24. J. Herrera, E. Huedo, and R. Montero, "Porting of Scientific Applications to Grid Computing on GridWay," Scientific Programming **13** (4), 317–331 (2005).