

Machine Learning in a Multi-Agent System for Distributed Computing Management

I V Bychkov¹, A G Feoktistov¹, I A Sidorov¹, A V Edelev², S A Gorsky¹ and R O Kostromin¹

¹Matrosov Institute for System Dynamics and Control Theory SB RAS, Lermontov St. 134, Irkutsk, Russia, 664033

²Melentiev Energy Systems Institute, Lermontov St. 130, Irkutsk, Russia, 664033

Abstract. We address the relevant problem of machine learning in a multi-agent system for distributed computing management. We propose a new approach to the agent learning in the system for managing job flows of scalable applications in a heterogeneous distributed computing environment, which includes high-performance computing clusters, as its main components. We manage parameter sweep applications that execute their jobs in a virtual machine environment. We use the specialized tools to implement such environment. In contrast to the known approaches, our approach is based on the integrated applying of methods for job classification and parameter adjustment of algorithms for functioning agents. Simulation modeling the environment allows eliciting the necessary knowledge for parameter adjustment. During the learning of agents, we use the expert knowledge of environment node administrators. An example of solving the complex practical problem that relates to studying energy development directions of Russia demonstrates advantages of the proposed approach.

1. Introduction

In the last decade, studies, connected with problems of strengthening subject orientation and intellectualizing technologies for development and use of a heterogeneous distributed computing environment (HDCE) that includes Grid-systems and cloud infrastructures, have become really relevant [1]. The progress in this direction is due to the need for the increasingly integrated use of heterogeneous environmental resources, as well as high-level support provided for end-users in the development and implementation of scalable applications.

A relevant approach to the distributed computing management in HDCE is applying multi-agent system (MAS) based on the use of market mechanisms while regulating the supply and demand of resources [2]. In such a system, an agent is a software entity that uses elements of artificial intelligence. Resource owners and their users endow it with rights and responsibilities to service and manage the computing process. Agents represent the interests of resources users and owners that often have conflicting criteria to define the computing process efficiency [3]. In processes of the executing the user jobs and coordinating their actions, they interact with each other. The agent coordination is based on their cooperation or competition. A selection of the cooperation or competition depends on their goals, roles, and mental properties.

The effectiveness of the agent work depends on the knowledge they use [4]. Stone [5] considers a wide range of basic capabilities and methods of the agent learning in systems with different

architectures. However, the agent learning remains an urgent problem in the tools of multi-agent computing management in practice, thus it reasonably requires development [6].

Various methods of machine learning have been developed [7, 8]. Usually, their purpose is an automatic improvement of decision-making quality over time in conditions of uncertainty in order to improve the efficiency of the controlled system [9]. Decision-making algorithms often depend on parameters that significantly affect the quality of management. A perspective direction of machine learning development is an integration of methods for an analysis of computational data and knowledge elicitation with the expert support of specialists in subject domains in the parametric adjustment process of these algorithms.

The paper addresses an approach to the agent learning based on a parametric adjustment of their algorithms for a job management in HDCE. In the computation management process, agents use the knowledge about both the problem specifics, which allow classifying jobs for their solving, and information about the environment, which ensure the rational distribution of the required resources.

2. Job management

In the paper, we consider HDCE that organized based on resources of the public access computer center “Irkutsk supercomputer center of the Siberian branch of the Russian Academy of Sciences” [10]. It supports two types of resources: dedicated (virtualized) and non-dedicated resources. The main components of the environment are high-performance clusters, the nodes of which differ in their computational characteristics.

MAS for jobs management in this environment includes agents for fulfillment the following operations:

- Problem formulation and problem-solving plan forming,
- Job classification,
- Creating a virtual community of agents representing environment resources,
- Parameter adjustment of agent functioning algorithms,
- Environment monitoring,
- Job dispatching in non-dedicated resources, etc.

These agents play the roles of the user agent, jobs-classification agent, agent-organizer, resource agent, parameter-adjustment agent, monitoring agent and agent-manager, respectively. Agents that represent environment resources can temporarily assume the role of agent-coordinator that regulates relationships of virtual community agents.

Representation of knowledge used by the agents is based on applying the conceptual model HDCE [11] that is a special case of a semantic network. In contrast to computational models of similar purpose (see, for example, [12]), such a model allows to carry out an interconnected description not only of algorithmic knowledge of subject areas for the solved problems, but also the knowledge about the hardware and software infrastructure of the environment and about administrative policies defined for its resources. The model includes the following knowledge components:

- Computational knowledge containing information on application modules for solving problems and system modules for computing planning, job-generating, resources allocation, monitoring of computational processes, dynamic decomposition of problems, and data preprocessing or postprocessing,
- Schematic knowledge comprising a set of objects (for example, parameters and operations) for describing the modular structure of the models and algorithms for the subject domain study,
- Production knowledge that defines the rules for applying operations and allows the applications end-users to select the best algorithms in the current computing situation,
- Infrastructure knowledge presented by the characteristics of hardware and software objects – nodes, communication channels, network devices, network topology and other structural elements, as well as information about their reliability,
- Administrative knowledge of policies against resources and users, including rules for the use of resources, rights and quotas for users and their jobs, and information about job management systems.

Figure 1 shows the job management scheme.

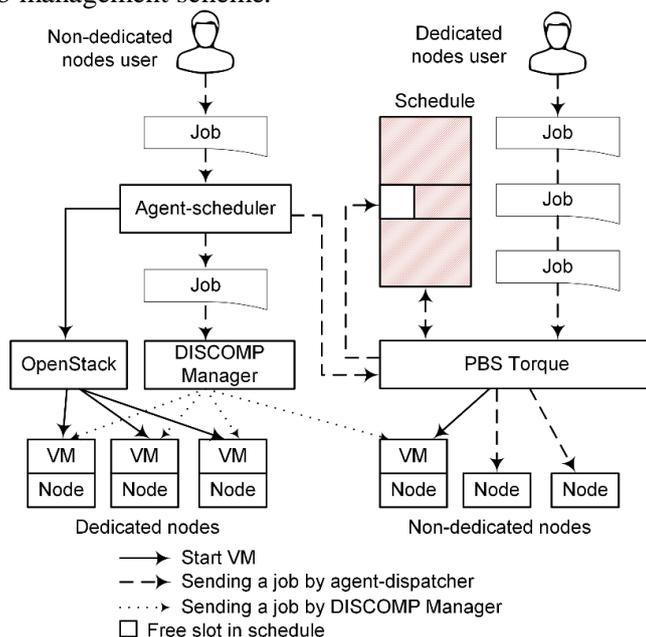


Figure 1. A scheme for multi-agent management of user jobs in a heterogeneous distributed computing environment using virtual machines.

Based on the problem formulation by the end-user of the application on the HDCE model, the user agent builds a set $P = \{p_1, p_2, \dots, p_k\}$ of problem-solving plans. Then the agent-organizer integrates resource agents into a virtual community through the knowledge about a conformity of the assigned module classes and available environment resources. The virtual community includes agents that represent the resources in which modules of a problem-solving plan can be run. Virtual community participants elect the agent-coordinator in the process of their local interactions based on the modified tree algorithm taking into account communication topology of an agent network.

We apply the tender of computational work to distribute modules of problem-solving plans between agents. It is based on the one-round model second-price Vickrey auction [13]. Within the tender process, each agent makes the offers for executing modules. The agent-coordinator determines the bidders-winners in the tender. Applying the computational work tender allows using additional criteria for the job execution quality in addition to their cost that is the single condition in auctions. These are such criteria as the problem-solving time, computing reliability, information safety and other restrictions. In the case of parameter sweep applications in which each module is executed multiple times with different values of their input parameters, the bidding is conducted for the right to process data variants using this module.

The agent-coordinator conducts the tender and determines the optimal problem-solving plan $p_{opt} \in P$. It also selects resource agents (bidders-winners) participating in the plan execution. Determining the plan and its executors is implemented using the multicriteria lexicographic method of a selection taking into account the given problem-solving efficiency criteria ordered on the degree of their importance.

Executing the modules of p_{opt} in the allocated resources appointed by the agents is carried out by tools of the DISCOMP toolkit in the asynchronous mode upon the data is ready [14]. The agent-manager runs the required number of virtual machines (VMs) using the OpenStack platform tools [15] and transfers the job to DISCOMP manager, which then sends the task to run the modules to the DISCOMP clients hosted in the VMs. In case a job queue occurs in the allocated resources, the agent-manager directs the task to run the VMs in the non-dedicated resources if there are free slots in the job execution schedule of PBS Torque [16].

3. Machine learning of agents

The process of the agent learning is based on the complex use of methods of conceptual modeling, job classification and parameter adjustment of the management system. Table 1 shows the agents with methods, tools, and subjects of their learning.

Table 1. Methods, means, and subjects of machine learning of agents.

Agent	Method	Tool	Subject
Agent for problem-solving planning	Conceptual modeling of the subject domain	Toolkit DISCOMP, the XML language extension	Application developer
	Formulating problems and criteria of the efficiency of their solving		Application end-user
Job classification agent	Attributive description of job classes	Job classification System	Environment administrator
Agent-organizer of agent virtual community	Matching job classes and resources	Job classification system	Environment node administrators
Resource agent	Parameter adjustment	Simulation modeling system	Parameter adjustment agent
Parameter adjustment agent	Configuration adjustment	MAS configuration adjustment system	Environment administrator
	Environment monitoring	Meta-monitoring system	Meta-monitoring agent
Meta-monitoring agent	Configuration adjustment	MAS configuration adjustment system	Environment administrator
Job management agent in non-dedicated resource	Matching job classes and resources	Job classification system	Environment administrator

The subject domain model, problem formulations, and criteria for their solving are described through the DISCOMP tools by the application developer and its users in XML. Figure 2 and Figure 3 show fragments of such a description.

```
<parameters>
  <param name='model' type='file' filename='model.txt'>
  <param name='model_list' type='filelist'
    pattern='model_element_%1.txt' />
  <param name='result_list' type='filelist'
    pattern='result_element_%1.txt' />
</parameters>
<modules>
  <module name='decompose'>
    <commands os='Linux'>
      <start>decompose.exe</start>
    </commands>
    <parameters>
      <input><param name='model' /></input>
      <output><param name='model_list' /></output>
    </parameters>
  </module>
  ...
</modules>
```

Figure 2. Subject domain model.

```
<process>
  <stage>
    <module name=' korrektiva_decompose' />
  </stage>
  <stage>
    <listmodule name='korrektiva_solver' />
  </stage>
  <stage>
    <module name=' korrektiva_analyse' />
  </stage>
</process>
```

Figure 3. Problem formulation.

In the job classification system [17], node administrators, based on their practical skills and experience, define the set $H = \{h_1, h_2, \dots, h_m\}$ of the possible job characteristics (problem-solving time, sizes of RAM and disk memory, number of nodes, processors and cores, module execution modes, etc.), and their domains (Figure 4). Next, they form the set $C = \{c_1, c_2, \dots, c_n\}$ of job classes that have characteristics from H . In case the characteristic domain is included in the concrete class c_i , it can be specialized (restricted). The formed classes are mapped to the most appropriate resources for executing jobs that belong to those classes.

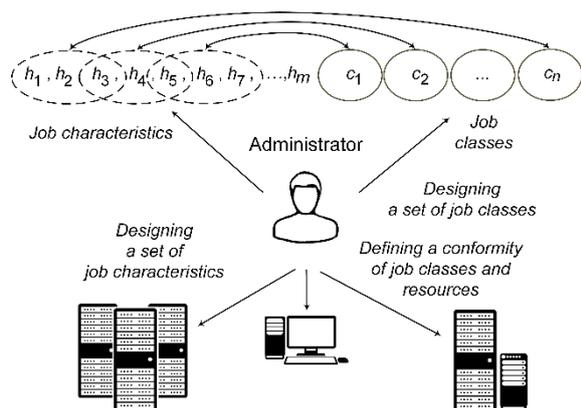


Figure 4. Designing the job classification system.

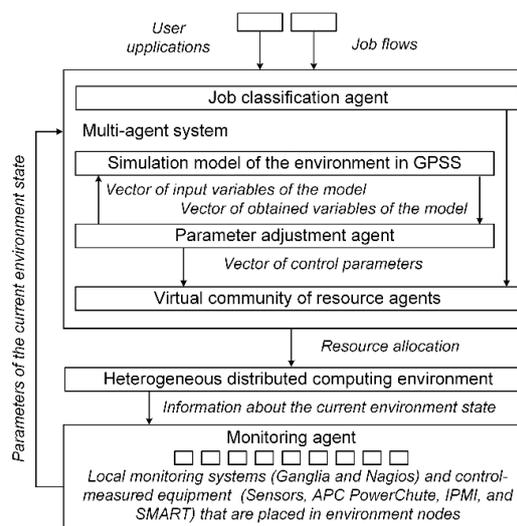


Figure 5. Parameter adjustment scheme.

The environment administrator sets the following configuration parameters:

- Parameter values that determine the intentions of resource agents to execute jobs of different classes,
- Lower and upper limits of the allowable deviation from the average resource load for agents of virtual communities,
- Amount of fines for the deviation from the average load (for resource agents),
- Composition and frequency of information collection, and data formats,
- Controlling and measuring means, and monitoring systems that will be used,
- Change limits of measured values and control actions applied when they are reached (for the monitoring agent);
- Permissible quotas on the number of jobs, their execution time and the number of nodes used,
- Characteristics of the slots in the PBS Torque system schedule in non-dedicated resources (for the jobs scheduling agent).

Figure 5 represents the parametric adjustment scheme of functioning algorithms of resource allocation agents. The agent-classifier identifies the job classes. The virtual community of resource agents is formed on the basis the matching classes to resources. These agents allocate resources. They use the tender of computational works and algorithms of their work.

The algorithms are determined by the vector of control parameters that provide agents with the selection of the optimal strategy of behavior. The parameter adjustment agent controls the parameters that reflect the values of the vector of input sweep variables of the HDCE simulation model. This vector corresponds to the optimal observed variables of the model that are calculated based on the parameter sweep computing. Multicriteria rules of the discrete selection ensure finding the optimal values [18].

The monitoring agent is designed to provide the HDCE subjects with up-to-date information on the loading of its resources, physical state of the equipment and engineering infrastructure devices [19]. In

contrast to other monitoring systems, an important feature of the monitoring agent use is the agent ability to analyze and apply control actions directly on the computing node where the agent operates. The monitoring agent collects, unifies, aggregates, and transmits data to the expert system for their analysis. In case critical events are detected, the necessary functions of the executive system of the agent are performed in order to apply control actions for automatic troubleshooting. At the same time, the administrator can pre-train monitoring agents taking into account the purpose of computing nodes and the jobs executed on them.

Evaluation of the successful agent learning is the class determination correctness by the agent-classifier and the resource use efficiency by their agents.

We apply an attributive description based on mandatory and optional sets of characteristics for a job specification [17]. If one or more optional characteristics are absented then an uncertainty can arise in the job classification. This uncertainty leads to the ambiguous in the class determination. The use of additional knowledge about the ranks, weights and computational history of job characteristics can significantly mitigate this uncertainty. The computational history is also used by the classifier-agent to evaluate its decisions. A set of characteristic functions is developed for recognition of job classes using different components of knowledge.

Figure 6 shows the results of classifying more than 80000 jobs of the real flow that were running on three clusters with different nodes. We compare the primary job class identification to the classification with using the additional knowledge. It is obvious that the class determination error in a percentage of the number of jobs with the uncertainty is significantly lower in the second case.

Evaluation of resource agent actions is regulated by the system of fines for deviation from the average resource load in their virtual community. Based on the analysis of the resource allocation results, agents can change the intent of executing different job classes.

Figure 7 shows the results of the average CPU load. These results show that the parameter adjustment of agents taking into account the specified job execution criteria significantly have been improved the processor load balancing.

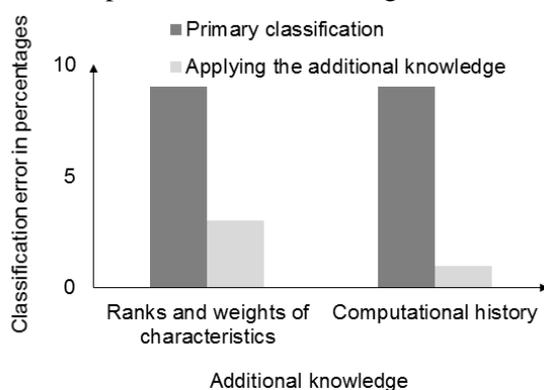


Figure 6. Job classification.

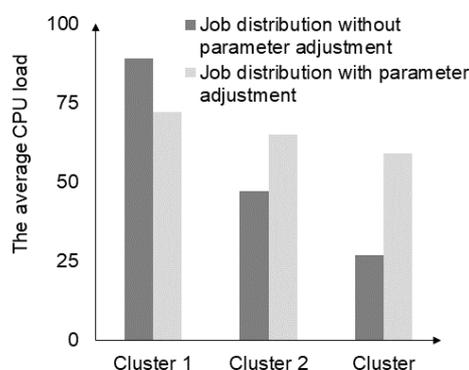


Figure 7. Load balancing.

4. Computational experiment

The example of solving a complex practical problem of determining the critical elements in technical infrastructure networks demonstrates the features and advantages of the proposed approach [20]. It consists of the study of failure sets, each of which represents a set of failed elements and has only one negative consequence of the impact on the system. The number n of simultaneously failed elements characterizes the failure set. The researcher selects the number n depending on the total number m of the system elements. For practical reasons, n has not exceeded 3 or 4 owing to the number of possible failures sets equal to $\frac{m!}{(m-n)!n!}$ is growing rapidly together with an increase of n .

To solve the problem, we developed a scalable application that supports the parameter sweep computing. The object of the study is the unified gas supply system of Russia. Its infrastructure contains 382 nodes, including 28 natural gas sources, 64 consumers, 24 underground gas storages and

266 key compressor stations, 486 arcs representing the main gas pipelines and outlets to distribution networks. We selected 415 arcs and 291 nodes (natural gas sources, underground gas storages, and key compressor stations) in this infrastructure. The selected 706 elements were calculated with $n=3$ and $n=4$. There is 58400320 and 10263856240 sets of failures for $n=3$ and $n=4$ correspondingly. The evaluated time to study all failure sets on one core of the Opteron 6276 Interlagos processor is 50 days if $n=3$ and more than 81 years if $n=4$. The evaluated time to study all failure sets on one core of the Intel Xeon E5-2695 processor is 14 days if $n=3$ and more than 32 years if $n=4$. These evaluations necessitate the use of high-performance computing.

To carry out an experiment, we create HDCE that includes the nodes of two segments of the HPC-cluster “Academician V. M. Matrosov”, which is a part of the Irkutsk supercomputer center. The environment nodes have the following characteristics:

- Two processors AMD Opteron 6276 Interlagos (16 cores, 2.3 GHz, 64 GB RAM) in the first segment,
- Two processors Intel Xeon E5-2695 v4 Broadwell (18 cores, 2.1 GHz, 128 GB RAM) in the second segment.

Existing quotas of the cluster resource allocation do not allow to carry out a fully computational experiment for $n=4$ owing to restrictions of the maximum number of the resources allocated to the user in one segment. A user cannot use more than 15 and 20 nodes with the maximum of a job service time that equal 20 and 10 days in the first and second segments respectively. If a user uses the resources of one segment, it can run jobs on the second segment only if there are free slots in the job schedule. Owing to the aforementioned restrictions, we select the following scheme of the computational experiment:

- Allocating 20 nodes for a period of 15 days in the schedule of the second segment (the maximum allowed time of the resources use within the existed quota),
- Using free slots in the schedule of the first segment (within the existed quota).

In the first segment, the schedule slots are determined in the presence of jobs that are waiting for its launch for the execution. The agent-manager interacts with the PBS Torque job queue manager to identify nodes that currently have free cores. The number of slots corresponds to the number of such nodes. The number of free node cores is the width of the slot. The period until the end of the nearest job is the slot duration. It cannot exceed 1 day.

The agent-manager predicts the time when the nearest job will be completed. Predicting the time of the resources release is based on the maximum requested job run-time using the coefficient reflecting the real execution time of user jobs taking into account the computational history.

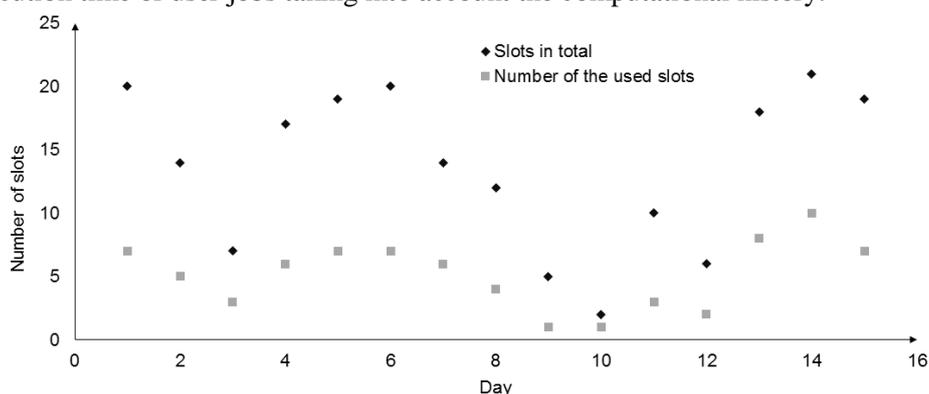


Figure 8. Free slots in the job schedule in the first segment.

Figure 8 shows the total number of slots in the job schedule in the first segment during the experiment and the number of slots used. We can see, that 62% of the slots were not used because of the possible computing inefficiency in them owing to high overheads for running and terminating VMs compared to the slot duration. Another reason not to use slots is the negative impact on the problem-solving processes of other users within the same node where the slot is.

Knowledge about the computing inefficiency in slots with specific characteristics is reflected in the job class description and applied by agents in the resource allocation process. The agent learning in the resource allocation process has increased the number of resources available for the experiment by 27 percent and completed the experiment in 15 days for $n = 4$.

5. Conclusions

We have considered a multi-agent system for distributed computing management in a heterogeneous distributed computing environment with virtualized resources. In contrast to the known multi-agent system, its functioning is based on the agents' complex use of the following knowledge:

- Computational knowledge of software modules for both the problem-solving in the subject problems and operating with system objects,
- Schematic knowledge of the modular structure of the model and algorithms,
- Productive knowledge to support decision-making on the selection of optimal algorithms depending on the environment state,
- Knowledge of the hardware and software infrastructure and administrative policies in its nodes.

This knowledge is represented in the form of a conceptual model that is a special case of the semantic network.

To support the agent learning, we have developed a new technology of the parameter adjustment of multi-agent algorithms for managing the heterogeneous distributed computing environment. An agent applies it to optimize the resource allocation when application user jobs are executed.

The proposed learning uses both the practical experience and skills of specialists in their subject domains (environment administrators, developers, and end-users of applications) and the knowledge elicited by agents. In contrast to the known approaches, selecting the control parameters of agent functioning algorithms within framework of the proposed agent learning are based on the integrated applying the job classification, matching classes to resources, meta-monitoring, and simulation modeling.

Thus, the developed technology allows performing detailed accounting the properties of distributed resources and characteristics of executed jobs, evaluating the current environment state, and predicting its evolution. Thereby, it provides a high degree of efficiency, reliability, and scalability of the computational process of solving large problems.

We have developed a scalable application to solve the important large-scale problem of studying development directions of the Russia energy sector from the standpoint of energy security in practice. We have also provided the intensive experiments to solve this problem based on the parameter sweep computing in HDCE. The experimental analysis confirms the effectiveness of multi-agent computing management and agent learning.

6. References

- [1] Talia D 2012 Clouds meet agents: Toward intelligent cloud services *IEEE Internet Comput.* **16** 78-81
- [2] Singh A, Juneja D and Malhotra M 2017 A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing *J. King Saud University Comput. Info. Sci.* **29** 19-28
- [3] Shyam G K and Manvi S S 2015 *Proc. of the 2015 IEEE Int. Advance Computing Conf.* 458-463
- [4] Talia D 2011 *Proc. of the 12th Workshop on Objects and Agents* **741** 2-6
- [5] Stone P and Veloso M 2000 Multiagent systems: A survey from a machine learning perspective *Auton. Robots.* **8** 345-383
- [6] Madni S H H, Latiff M S A and Coulibaly Y 2017 Recent advancements in resource allocation techniques for cloud computing environment: a systematic review *Cluster Comput.* **20** 2489-2533
- [7] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Berlin, Heidelberg: Springer)

- [8] Murphy K P 2012 *Machine Learning: a Probabilistic Perspective* (Cambridge: MIT Press)
- [9] Jordan M I and Mitchell T M 2015 Machine learning: Trends, perspectives, and prospects *science* **349** 255-260
- [10] Access mode: <http://hpc.icc.ru>
- [11] Bychkov I, Oparin G, Tchernykh A, Feoktistov A, Bogdanova V and Gorsky S 2017 Conceptual model of problem-oriented heterogeneous distributed computing environment with multi-agent management *Procedia Comput. Sci.* **103** 162-167
- [12] Oparin G A, Feoktistov A G and Feoktistov D G 1996 Combined abstract-program execution in the Saturn instrumental complex *Autom. Control Comp. S.* **30** 57-61
- [13] Vickrey W 1961 Counterspeculation, auctions, and competitive sealed tenders *J. Finance* **16** 8-37
- [14] Edelev A V and Sidorov I A 2017 Combinatorial modeling approach to find rational ways of energy development with regard to energy security requirements *Lecture Notes Comp. Sci.* **10187** 310-317
- [15] Bumgardner V K 2016 *OpenStack in Action* (Manning Publications)
- [16] Access mode: <http://www.adaptivecomputing.com/products/open-source/torque/>
- [17] Feoktistov A, Tchernykh A, Kostromin R and Gorsky S 2017 Knowledge Elicitation in Multi-Agent System for Distributed Computing Management *Proc. of the 40th Int. Convention on information and communication technology, electronics and microelectronics* 1350-1355
- [18] Bychkov I V, Oparin G A, Feoktistov A G, Sidorov I A, Bogdanova V G and Gorsky S A 2016 Multiagent control of computational systems on the basis of meta-monitoring and imitational simulation *Optoelectron. Instrum. Data Process.* **52** 107-112
- [19] Bychkov I, Oparin G, Novopashin A and Sidorov I 2015 Agent-based approach to monitoring and control of distributed computing environment *Lecture Notes Comp. Sci.* **9251** 253-257
- [20] Jonsson H, Johansson J and Johansson H 2008 Identifying critical components in technical infrastructure networks *Proc. Inst. Mech. Eng. O J. Risk. Reliab.* **222** 235-243

Acknowledgments

The study is partially supported by Russian Foundation of Basic Research, projects no. 16-07-00931-a and no. 18-07-01224-a, and Presidium RAS, program no. 30, project "Methods, algorithms and tools for the decentralized group solving of problems in computing and control systems".