Subject-oriented computing environment for solving large-scale problems of energy security research

I V Bychkov¹, G A Oparin¹, A G Feoktistov¹, I A Sidorov¹, S A Gorsky¹, R O Kostromin¹ and A V Edelev²

¹Matrosov Institute for System Dynamics and Control Theory SB RAS, Lermontov Street, 134, Irkutsk, Russia, 664033
²Melentiev Energy Systems Institute SB RA, Lermontov Street, 130, Irkutsk, Russia, 664033
e-mail: agf@icc.ru

Abstract. We represent a new approach to the development and use of a subject-oriented heterogeneous distributed computing environment for solving large-scale problems of energy security research. Both the development of scalable applications (distributed applied software packages) and the computing resources integration are conducted using the Orlando Tools framework. It supports the collaborative application development based on the continuous integration of their software. Computation management tools in Orlando Tools are integrated with a multi-agent system. We propose special multi-agent tools for the heterogeneous resource allocation in the environment. These tools apply knowledge about the subject domain specifics of solved-problems and computing environment characteristics. The effectiveness of such management is highlighted by the results of solving practical problems of energy security research.

1. Introduction

Energy security of a country is characterized as the state of protection of its citizens, society and the economy from a shortage in ensuring their reasonable energy needs. Energy security research is divided into strategic and tactical. A goal of strategic research is to understand the reasons for the emergence and implementation of threats to energy security in the form of emergencies. The strategic research orients to the development of the key directions of emergencies prevention. It supposes conducting the indicative analysis of modern and retrospective information about external conditions and the state of the energy sector [1].

Mathematical methods and models of the tactical research make it possible to assess the scale of the energy resource shortage, determine possible measures to counteract the emergency development, and outline ways to restore the needed volume of energy as soon as possible [2]. A goal of the aggregate research technology is supporting experts in creating an integrated view of the energy security level of regions and the country on the whole. The technology should help experts in decision-making, developing recommendations, and implementing measures to ensure the energy security based on an analysis of all possible factors including environmental, social, political, and other requirements and restrictions. The tactical energy security research is based on a combinatorial
analysis of a large number of combinations of emergencies and energy development directions. Conducting such an analysis reasonably requires the use of high-performance computing.

Nowadays, basic software tools that implement technologies for organizing computations in high-performance computing systems provide a potential basis for the mass development of parallel and distributed applications. However, an analysis of their applying in practice reveals the following important problems:

- Difficulty of training and applying these tools by end-users that significantly narrows their community,
- Insufficient support for taking into account the problem specifics,
- Lack of universal approaches to resource integration,
- Weak integrability of applications on data and computation management.

In addition, there is the problem of matching the problem-solving criteria (time, cost, reliability, etc.) and resource owner preferences (resource utilization, load balancing, power consumption, etc.) in the process of distributed computing management [3].

The above-listed problems are especially actualized when solving problems in a heterogeneous distributed computing environment. Its main components are clusters including hybrid clusters with heterogeneous nodes, high-performance servers, data storage systems, personal computers (PCs), and other computational complements. Besides, users still use PC-clusters [4]. To solve the problem in the environment, the user has to create a job for a local resource manager used in environment nodes.

The environment created on the basis of the resources of a public access computer center differs significantly in its characteristics from homogeneous systems represented, for example, by supercomputers with a multiprocessor configuration or reconfigurable computing systems. Within the center, local users of clusters and global users of the environment solve their problems.

The paper address the main aspects of the new approach to the development and use of the subject-oriented environment for solving problems of energy security research. An original multi-agent model of the environment resources allocation is proposed. Using the model, agents interact with each other in the resource allocation process applying knowledge about the problem-solving specifics of the subject domain and environment resources characteristics. The model provides high efficiency in managing flows of jobs generated by scalable applications. The development and use of such applications are conducted using the Orlando Tools framework [5].

The rest of the paper is organized as follows. The second section provides a brief overview of research related to supporting parallel and distributed computing in solving large-scale problems in various fields of energy based on combinatorial modeling. Section 3 describes an experimental subject-oriented environment for solving problems of energy security research. Section 4 proposes a multi-agent approach to distributed computing management in the environment. A new subsystem for continuous integration of modules at the Orlando Tools framework for developing distributed applied software packages is suggested in Section 5. Section 6 represents the experimental analysis results in solving the practice problems. Finally, Section 7 concludes with the main results of our study.

2. Related work

In the last decade, research related to the development and use of specialized environments for solving large-scale scientific and applied problems in various subject domains of energy based on combinatorial modeling is actively evolving. The relevance of such research is due to the fact that the high-performance computing use provides the earliest possible breakthrough results and competitive advantages in scientific, industrial, economic and other spheres of human activity.

Combinatorial modeling in the research of energy security is often characterized by the large computational difficulty, need for complex studies (solving a number of interrelated classes of problems), and high intensity of applied software changes [6]. At present, parallel and distributed computations are actively used to implement combinatorial modeling [7–11].

However, as shown in [12], when moving from parallel computing systems with shared memory to distributed environments, the computation speedup and resource utilization efficiency are significantly
reduced. This is especially evident when using heterogeneous resources. Applying the traditional Grid and cloud computing management tools does not improve this situation.

In contrast to the aforementioned environments, the proposed environment for solving problems of energy security ensures the development and use of distributed applied software packages and their integration through the data and computation management. It integrates Grid and cloud computing in the problem-solving process under multi-agent management. In addition, a selection of rational resource configuration for conducting experiments of various scales is supported in the environment.

Usually, the effective resource allocation for executing jobs of scalable applications is a non-trivial problem for the well-known computing management systems [13]. Providing a subject orientation of the environment [14], applying multi-agent technologies [15], and supporting the continuous integration of applied software [16] can significantly improve the efficiency of computing management. The research related to the integration of Grid and cloud computing into the hybrid computation model is also extremely relevant [17]. The combination of the aforementioned capabilities is a great challenge. An analysis of the state-of-the-art in the field of developing subject-oriented distributed computing environments shows that there is no common approach to solve this challenge. In Table 1, we represent the degree of supporting the above-listed capabilities within well-known approaches to creating such environments, including our approach. The marker ‘+/-‘ means that some systems have the considered capabilities and others do not. Obviously, our approach supports all needed capabilities in comparison with other approaches.

| Approach | Subject orientation | Multi-agent management | Continuous integration | Hybrid computation model |
|----------|---------------------|------------------------|------------------------|-------------------------|
| Applying various systems for developing and managing workflows [18, 19] | + | +/- | - | +/- |
| Voluntary computing with the help of the BOINC system [20] | + | - | - | +/- |
| Using the X-COM system [21] | + | - | - | - |
| Applying the Orlando Tools framework | + | + | + | + |

3. Experimental subject-oriented environment for solving problems of the energy security research

The environment is created using the Orlando Tools framework. It included in the basic system software of the Irkutsk Supercomputer Center (ISC) of the Siberian Branch of the Russian Academy of Sciences [22]. The process of developing a distributed applied software package in Orlando Tools includes the following three main stages:

- Structural analysis of the subject domain,
- Development of the applied content of the package (application software modules),
- Formation of the computation infrastructure.

At the stage of the structural analysis of the subject domain, the basic concepts are unified using the methods of conceptual modeling. These methods support the following operations:

- Designing energy system schemes,
- Determining the relations between energy systems,
- Developing mathematical energy system models,
- Matching energy system models,
- Defining database structures and file formats,
- Preparing, decomposing, and aggregating the subject data.

As a result, a model of the subject domain of the package is constructed. It describes problem-solving processes in terms of parameters (meaningful values of the subject domain) and abstract operations on the field of parameters. At the second stage, software modules are developed. They
implement the operations. At the final stage, the resources that can be used to execute the modules are determined. These resources are registered with Orlando Tools.

Orlando Tools provides integration of developed packages by using already created fragments of conceptual models of packages including parameters associated with existing databases and operations related to package modules.

Figure 1 shows the architecture of a subject-oriented environment for solving problems of the energy security research. Depending on the problem dimension, the computation infrastructure of the distributed applied software package may include PCs, servers, clusters, Grid systems and cloud platforms. Their integration into a unified integrated computing environment provides the ability to select the required hardware and software configuration of resources. In addition, it ensures the effectiveness of preparing and conducting experiments on a various scale.

The integration of computing resources is realized by the specialized Orlando Tools server. It also implements the system part of the developed packages and user interface for accessing package components. The system part includes the following subsystems:

- Subsystem for administration and configuration of computation resources,
- Designer of the conceptual model of the environment,
- Computation planners,
- Interpreters of problem-solving schemes,
- Subsystem for continuous integration of applied modules of packages.

The system of continuous integration is designed for debugging and testing modules, managing their versions both in the libraries of package modules and external repositories, synthesis of problem-solving schemes, and they testing in the automated mode.

The knowledge base contains the knowledge components of the conceptual model of the environment. Issues related to the construction and applying of the model are discussed in detail in [23, 24]. The calculated database stores the information necessary for conducting experiments and writes problem-solving results. Connection of remote information and computation resources is performed using the API to support Grid and web-services.

The Orlando Tools means for computation management at the application level are integrated with the multi-agent system (MAS) [25]. It is used as an intelligent superstructure over local resource.
managers located at the environment nodes. We apply MAS in order to optimize resource allocation. Users of the environment with the help of agents form computational jobs for solving their problems. Jobs describe information about the required system hardware and software of environment resources, applied software, input and output data, criteria of problem-solving, and other necessary data.

4. Multi-agent management of distributed computing

We apply MAS to implement computation management at the environment level [25]. MAS belongs to the class of meta-schedulers. It has a hierarchical structure and functions on the basis of the principles of self-organization. At each level of the system, agents have different rights and duties delegated to them by the owners and users of resources. They perform functions appropriate to their status. Agents unite in virtual communities. Computation management is implemented based on local agent interactions.

All jobs sent to the environment are classified according to their characteristics. The resource allocation is conducted with the help of economic mechanisms for regulating their supply and demand in the framework of the tender for computational work. The tender is implemented on the basis of the auction Vickrey [26]. In the auction model, bidders compete only by one criterion - the price of the lot. Within the framework of the tender, various conditions for performing the work can be determined additionally. In the process of problem-solving, the dynamic re-allocation of computational work (jobs) is possible. The computation management scheme is represented in figure 2.

![Figure 2. Computation management scheme.](image)

We consider the environment as a control object. External disturbances for the control object are the flow $w_1$ of user jobs and flow $w_2$ of local users of clusters. The distribution results $d_1$ and $d_2$ for these flows between clusters are control actions of MAS and local users to the control object. Corrective action for the environment from its nodes administrators is the vector $r_1$ of parameters of administrative policies for clusters. Resource allocation agents intercept the job flow $w_1$ for the purpose of more detailed adjustment of the requirements for the computation system contained in the jobs. The flow $w_1$ is transformed into the flow $w_1^*$. The distribution $d_2$ of the flow $w_2$ is determined by local users.

Information about the environment node characteristics and computational processes is collected by the special meta-monitoring system using control and measuring equipment, and local resource managers placed in the nodes [27]. It is represented by the file data structures $a$ and $b$.

The structure $b$ is defined by the abstract connection $F$:

$$b = F \left( a, r_1, w_1^*, d_1, w_2, d_2 \right).$$

Different elements of the structure $b$ can match various mapping (functional, statistical, ambiguous, etc.). The connections $Q$ and $H$ have the similar character as the connection $F$. 
The collected information (the vector $c_1$ of aggregated parameters of the environment functioning) is transferred to the agent for parameter adjustment according to its request. The agent for parameter adjustment sends requests to the meta-monitoring system with a period $T_1$ of discreteness. The vector $c_2$, formed on the basis of the vector $c_1$, includes information required for resource allocation agents.

Corrective action for the environment from its administrator is the vector $r_2$ of parameters of administrative policies for the environment on the whole. At the given time moments, the agent for parameter adjustment predicts changing environment functioning parameters for a certain period using the simulation modeling system [27]. In decision-making, it based on the information provided by vectors $c_1$ and $r_2$, and simulation modeling results.

The agent for parameter adjustment forms the vector $u$ of control actions. This vector is used for the parameter adjustment of the operation of resource allocation agents. Elements of the vector $u$ determine the following values: restrictions on the node component utilization, bonuses for meeting these restrictions, fines for their excess, priorities of the job classes, and degree of a desire to perform jobs of certain classes. The formed vector $u$ is transferred to each resource allocation agents. The agent for parameter adjustment initiates the simulation modeling of the environment functioning with the period $T_3 > T_1$ of discreteness. Input and observed variables of the simulation model are represented by the vectors $x$ and $y$.

Figure 3. Continuous integration process in Orlando Tools.

5. Continuous integration
An important characteristic of distributed applied software packages is a tendency to the change of algorithmic knowledge implemented by modules that can be developed and modified by different developers and end-users. In this regard, we merge a technology of developing the packages with the continuous integration of software using conceptual modeling for describing information processes. A continuous integration process in Orlando Tools involves the following operations:

- Development and modification of modules,
- Compiling, debugging, and testing modules and problem-solving schemes,
- Placing modules into repositories,
- Installing modules on resources.

This process showed in figure 3. We implement the version control of the module source code using the Git system. Next, we apply the GitLab system for repository management. It provides
programming code analysis, error tracking, module versions testing, etc. JFrog Artifactory is used for work with binary versions of modules. A combination of the external systems is based on the conceptual model of Orlando Tools. The model specifies all processes of continuous integration.

6. Experimental analysis
We use the experimental environment for solving the following problems:

1) Analysis of the energy development of Vietnam taking into account the energy security requirements,
2) Determination of critical elements of energy systems.

Their mathematical formulations are given respectively in [28, 2].

![Figure 4. Problem-solving schemes s1 (a) and s2 (b) in the graphical editor of Orlando Tools.](image)

The problem-solving scheme $s_1$ and computational model of the problem 1 are shown in figure 4a and figure 5a respectively. The computational model is represented by an oriented acyclic bipartite graph. This graph includes two disjoint sets of vertices: parameters and operations.

The operation $o_1$ creates an energy development graph. Then, the operation $o_2$ decomposes the graph into a set of subgraphs in parallel. Next, the obtained subgraphs are parallel evaluated by instances of the operation $o_3$. Finally, the operation $o_4$ integrates computation results and creates an optimal trajectory of energy development.

The parameters $p_5$ and $p_6$ are parallel lists of the elements $p_{5,1}, p_{5,2}, \ldots, p_{5,n}$ and $p_{6,1}, p_{6,2}, \ldots, p_{6,n}$ respectively. The operations $o_2$ and $o_3$ are represented by sets of their instances that one-to-one match elements of the parameters $p_5$ and $p_6$.

Figure 4b and figure 5b show the problem-solving scheme and computational model of the problem 2 respectively. The operation $o_5$ forms the parameter $p_{22}$ with failure sets of the specific size $p_{16}$, which determined by the operation $o_8$. The parameter $p_{22}$ is formed based on the time moment $p_{10}$, at which the energy system network data is retrieved, resource list $p_{12}$, failed elements list $p_{14}$ of an
energy system, and others parameters. Then, the operation \( o_{10} \) evaluates the consequences of element failures (parameter \( p_{21} \)) on the energy system \( p_{20} \) and writes the evaluation results into the distributed database \( p_{8} \). The parameters \( p_{15}, p_{17} \) and \( p_{18} \) defines the extra number of records to keep the consequences data, record number, and variable block size in the database. They are calculated by the operations \( o_{1} \) and \( o_{6} \) respectively based on the energy system scheme \( p_{11} \). The operation \( o_{5} \) determines the number \( p_{9} \) of the last record in the database. The parameter \( p_{23} \) is a composite parameter. It includes the parameters \( p_{10}, p_{11}, p_{13}, p_{14}, p_{17} - p_{20} \). The criticality of energy system elements is determined by the parameter \( p_{21} \). The more consequences of element failures mean the more its criticality.

![Figure 5. Computational model of the problem 1 (a) and problem 2 (b).](image)

Within the environment, dedicated and non-dedicated resources of ISC and Melentiev Energy Systems Institute (ESI) of the Siberian Branch of the Russian Academy of Sciences are used. Their characteristics are listed in Table 2. To execute modules that implement operations of the problem-solving schemes, virtual machines are launched on dedicated resources. In contrast to traditional meta-schedulers, the special hypervisor shell implemented in MAS provides the launch of jobs for performing additional virtual machines that are started on non-dedicated resources if there are free slots in their schedules. Such slots must meet the constraints imposed by administrative policies at environment nodes. All virtual machines running on dedicated and non-dedicated resources are combined by the Orlando Tools scheduler into a single virtual environment.

Figure 6 shows the average processor utilization of dedicated and non-dedicated resources. For the both problems, the average processor utilization of non-dedicated resources is 2-3% less than the average processor utilization of dedicated resources. This is owing to the additional overhead to launch and stop virtual machines in slots of non-dedicated resources. The real time \( t_{1} \) of solving problems on both types of resources with multi-agent management, and estimates \( t_{2} \) and \( t_{3} \) the time to solve these problems on the same resources without multi-agent control and only on dedicated
resources under the OpenStack management are shown in figure 7. It should be noted that traditional meta-schedulers cannot usually ensure the sharing of dedicated and non-dedicated resources. The results shown in figure 7 demonstrate the effectiveness of multi-agent management.

Table 2. Heterogeneous resources.

| Problem | Dedicated resources                                                                 | Non-dedicated resources                                                                 |
|---------|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| 1       | 5 nodes (2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 64 GB RAM) of ISC                 | Nodes (2x18 cores CPU Intel Xeon X5670, 2.1 GHz, 128 GB RAM) with free slots in the schedule of ISC; PCs of ESI |
| 2       | 10 nodes (2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 64 GB of RAM), 10 nodes (2x18 cores CPU Intel Xeon X5670, 2.1 GHz, 128 GB RAM) of ISC | Nodes (2x18 cores CPU Intel Xeon X5670, 2.1 GHz, 128 GB RAM) with free slots in the schedule of ISC; PCs of ESI |

The computational complexity of the considered problems directly depends on the number of states of the studied energy systems. An increase in the number of resources of the experimental environment will allow solving more complex problems in it. Figure 8 and figure 9 show the speedup and efficiency of solving such problems in the environment with 9110 cores obtained through simulation modeling.

Figure 6. Average processor load for dedicated and non-dedicated nodes.

Figure 7. Problem-solving time.

Figure 8. Speedup.

Figure 9. Efficiency.

Figure 10. Average time spent by a package developer.
Figure 10 demonstrates the average time that a package developer spent on the one module modification with non-automated applying the external tools of continuous integration and using the automated continuous integration in Orlando Tools. In the first case, we used the results obtained in developing the similar packages [29, 30]. Their developers made 30-40 module modifications in a one week. In the second case, we measured developer operation time. In comparison with non-automated applying the external systems, continuous integration in Orlando Tools enables the developer to significantly decrease the time at the stages of the modification modules or the package on the whole, planning and executing tests, and verification of their results.

7. Conclusions
The paper addresses the creation and applying subject-oriented computing environments. Generally, the efficient resource allocation for executing jobs of scalable applications is a non-trivial problem for the well-known computation management systems in such environments.

To this end, we propose a new approach to the development and use of scalable scientific applications (distributed applied software packages) in a heterogeneous distributed computing environment. It is based on the integrated applying of methods and tools of artificial intelligence, knowledge engineering, conceptual programming, distributed and parallel computing, and multi-agent technologies. The developed tools of the multi-agent management of distributed computing provide the matching of preferences of resource owners and quality criteria for solving problems that are set by environment users.

In addition, our approach supports interdisciplinary studies implementation and the collaborative application development based on the continuous integration of their software.

The advantages of the proposed approach are demonstrated in solving large-scale problems of energy security research. The experimental analysis showed the high scalability of the developed applications for solving the above-listed problems and resource use efficiency in the environment.

8. References
[1] Rabchuk V and Senderov S 2018 Analysis of the scale of the implementation of the energy security threats in Russia up to 2030 and the main directions for the neutralization of these threats E3S Web of Conferences 69 02011
[2] Vorobev S, Senderov S and Edel'ev A 2018 Search of critically important combinations of objects of the gas industry from the positions of the system operability E3S Web of Conferences 58 03002
[3] Toporkov V, Yemelyanov D and Toporkova A 2017 Anticipation scheduling in grid with stakeholders preferences Communications in Computer and Information Science 793 482-493
[4] Xue X, Xiang F and Wang H 2019 A parallel fusion method of remote sensing image based on NSCT Computer Optics 43(1) 123-131 DOI: 10.18287/2412-6179-2019-43-1-123-131
[5] Feoktistov A, Gorsky S, Sidorov I, Kostromin R, Edel'ev A and Massel L 2019 Orlando tools: Energy research application development through convergence of grid and cloud computing Communications in Computer and Information Science 965 289-300
[6] Huang G, Wang J, Chen C, Guo C and Zhu B 2017 System resilience enhancement: Smart grid and beyond Front. Eng. Manag. 4(3) 271-282
[7] Li X, Wei J, Li T, Wang G and Yeh W W G 2014 A parallel dynamic programming algorithm for multi-reservoir system optimization Adv. Water Resour. 67 1-15
[8] Cheng C, Wang S, Chau K W and Wu X 2014 Parallel discrete differential dynamic programming for multireservoir operation Environ. Model. Softw. 57 152-164
[9] Zhang Y, Jiang Z, Ji C and Sun P 2015 Contrastive analysis of three parallel modes in multi-dimensional dynamic programming and its application in cascade reservoirs operation J. Hydrol. 529 22-34
[10] Feng Z, Niu W, Cheng C and Wu X 2017 Peak operation of hydropower system with parallel technique and progressive optimality algorithm Int. J. Electr. Power Energy Syst. 94 267-275
[11] Feng Z, Niu W and Cheng C 2018 Optimization of hydropower reservoirs operation balancing generation benefit and ecological requirement with parallel multi-objective genetic algorithm
Energy 153 706-718

[12] Sharma T, Glynn J, Panos E, Deane P, Gargiulo M, Rogan F and Gallachóir B 2019 High performance computing for energy system optimization models: Enhancing the energy policy tool kit Energy Policy 128 66-74

[13] Tao J, Kolodziej J, Ranjan R, Prakash Jayaraman P and Buyya R 2015 A note on new trends in data-aware scheduling and resource provisioning in modern HPC systems Futur. Gener. Comput. Syst. 51(C) 45-46

[14] Sokolinsky L B and Shamakina A V 2016 Methods of resource management in problem-oriented computing environment Program. Comput. Softw. 42(3) 17-26

[15] Kalyaev A, Kalyaev I, Khisamutdinov M, Korovin I and Schaefer G 2016 An effective algorithm for multiagent dispatching of resources in heterogeneous cloud environments 5th International Conference on Informatics, Electronics and Vision 1140-1142

[16] Duvall P M, Matyas S and Glover A 2007 Continuous Integration: Improving Software Quality and Reducing Risk (Addison Wesley Professional)

[17] Mateescu G, Gentzsch W and Ribbens C J 2011 Where HPC meets grid and Cloud Computing Futur. Gener. Comput. Syst. 27(5) 440-453

[18] Yu J and Buyya R 2005 A Taxonomy of Workflow Management Systems for Grid Computing J. Grid Comput. 3 171-200

[19] Talia D 2013 Workflow Systems for Science: Concepts and Tools ISRN Software Eng. 2013 DOI: 10.1155/2013/404525

[20] Anderson D 2004 Boinc: A System for Public-Resource Computing and Storage 5th IEEE/ACM International Workshop on Grid Computing 4-10

[21] Voevodin V V 2007 The Solution of Large Problems in Distributed Computational Media Autom. Remote Control 68(5) 773-786

[22] Irkutsk Supercomputer Center (ISC) URL: http://hpc.icc.ru

[23] 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

[24] Bychkov I, Feoktistov A, Kostromin R, Sidorov I, Edelev A and Gorsky S 2018 Machine Learning in a Multi-Agent System for Distributed Computing Management CEUR-WS Proceedings 2212 89-97

[25] Bychkov I V, Oparin G A, Feoktistov A G, Bogdanova V G and Pashinin A A 2015 Service-oriented multiagent control of distributed computations Autom. Remote Control 76(11) 2000-2010

[26] Vickrey W 1961 Counterspeculation, Auctions, and Competitive Sealed Tenders J. Finance 16(1) 8-37

[27] Bychkov I V, Oparin G A, Feoktistov A G, Sidorov I A, Bogdanov 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

[28] Edelev A, Zorkaltsev V, Gorsky S, Van Binh D and Nam N H 2017 The combinatorial modelling approach to study sustainable energy development of Vietnam Commun. Comput. Inf. Sci. 793 207-218

[29] Edelev A, Sidorov I and Feoktistov A 2018 Heterogeneous Distributed Computing Environment for Vulnerability Analysis of Energy Critical Infrastructures Adv. Intell. Syst. Res. 158 37-42

[30] Feoktistov A, Gorsky S, Sidorov I, Kostromin R, Edelev A and Massel L 2019 Orlando tools: Energy research application development through convergence of grid and cloud computing Commun. Comput. Inf. Sci. 965 289-300

Acknowledgments
The study was fulfilled within the scientific projects III.17.5.1 and IV.38.1.1 of the basic research program of the SB RAS. It was supported in part by the Russian Foundation for Basic Research, project no. 19-07-00097-a.