

Methods and tools for evaluating the reliability of information and computation processes in Grid and cloud systems

A G Feoktistov¹, I A Sidorov¹, R O Kostromin¹, G A Oparin¹ and
O Yu Basharina²

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

²Irkutsk State University, Karl Marx St. 1, Irkutsk, Russia, 664003

agf@icc.ru

Abstract. The paper addresses the relevant issue of ensuring the reliability of solving large scientific and applied problems in computing environments that integrate Grid and cloud computing. The main reliability parameter is the probability of successful problem-solving in a computing environment with the following specified quality criteria: efficiency of the allocated resources use, and time, deadline or cost of executing jobs. We propose a new technology for testing and evaluating the reliability of functioning of problem-oriented heterogeneous distributed computing environments. It integrates models, representing different layers of knowledge about the environments, and special tools that automate a study of these environments. Applying such technology provides an increase in the reliability and efficiency of heterogeneous distributed computing environments by parametric adjusting of local resources managers installed in the environment nodes. Their adjustment is implemented on the base of the results of testing and evaluating obtained with the use of complex (conceptual, simulation, and semi-natural) modeling and meta-monitoring of computational resources.

1. Introduction

Today, in the development and application of high-performance computing, there are tendencies towards the organization of problem-oriented heterogeneous distributed computing environments, including computing Grid-systems for various purposes. Usually, such environments have a number of properties that significantly complicate their constructing, applying and studying. Among them are the following properties:

- Organizational and functional heterogeneity, dynamism and incompleteness of the description of integrated resources,
- Variety of the spectrum of problems solved with the help of these resources,
- The presence of various categories of users who pursue their goals and objectives of operating a computing system.

From the point of view of the study of heterogeneous distributed computing environments, the most important issues at present are their testing and reliability evaluation. Traditional management tools for distributed computing do not fully solve the above issues. In this case, the development of special tools is required to ensure the possibility of studying the reliability of the functioning of problem-oriented heterogeneous distributed computing environments.

Currently, there is a wide range of methods and tools for studying various aspects of the functioning of distributed computing systems (WorkflowSim [1], GridSim [2], and other systems [3]).

However, they are highly specialized. Usually, such tools provide a study of the computing system, including its software or hardware reliability, by analytic, simulation, or semi-natural modeling with a weak reference to the features of subject domains of solved problems.

In this regard, we propose a new approach to the development of tools for testing and evaluating the reliability of heterogeneous distributed computing environments.

The paper is structured as follows. Section 2 briefly reviews related works. Section 3 describes the considered heterogeneous distributed computing environment. In Section 4, we propose a conceptual model of the considered environment. The next two sections respectively represent our approach to simulation and semi-natural modeling of job flow executing in the environment. Section 7 provides the main aspects of technology for testing and evaluating the reliability of the environment. Section 8 concludes our study.

2. Related work

A special class of software tools is used to model distributed computing environments. Depending on the imitation method of the environment processes functioning, these tools are divided into the following two classes:

- Emulators,
- Simulators.

An emulator is a tool for imitating the behavior of a real device or program in real time. It replaces the properties and functions of the original, performing a real job. For example, emulators are the following systems: MicroGrid [4], Grid eXplorer [5], and AGNES [6].

A simulator is a software that enables us to simulate real systems, reflecting a part of real phenomena and properties in a virtual environment. It only models these properties and functions without real functioning. For example, the following systems are simulators: Bricks [7], CloudSim [8], OptorSim [9], and SimGrid [10]. These systems provide an opportunity to obtain statistical data on the most important characteristics of the modeled environment: the use of disk resources, channel capacity, the probability of data loss, and other properties of the environment functioning.

The above-listed systems have different features and capabilities for modeling the environment. There are a number of significant drawbacks that complicate the use of these systems for the simulation of subject-oriented distributed computing environments. Among them are the following disadvantages:

- Lack of support for conceptual modeling,
- Insufficient accounting the specifics of problem domains for solved problems,
- Weak typification of computational jobs,
- Limitations of simulated architectures of distributed computing environments,
- The absence of support for complex modeling (joint analytical, imitation, and semi-natural modeling).

In addition, the aforementioned systems are focused on the study of certain aspects in the functioning of distributed computing environments. Their integrated use is complicated by the need to unify and interpret data presented in various formats. There is no unified user-interface to these systems.

3. Heterogeneous distributed computing environment

In the article, we consider a heterogeneous distributed computing environment that is characterized by a number of features. Figure 1 shows a scheme that reflects the interaction of the main subjects and objects of the environment.

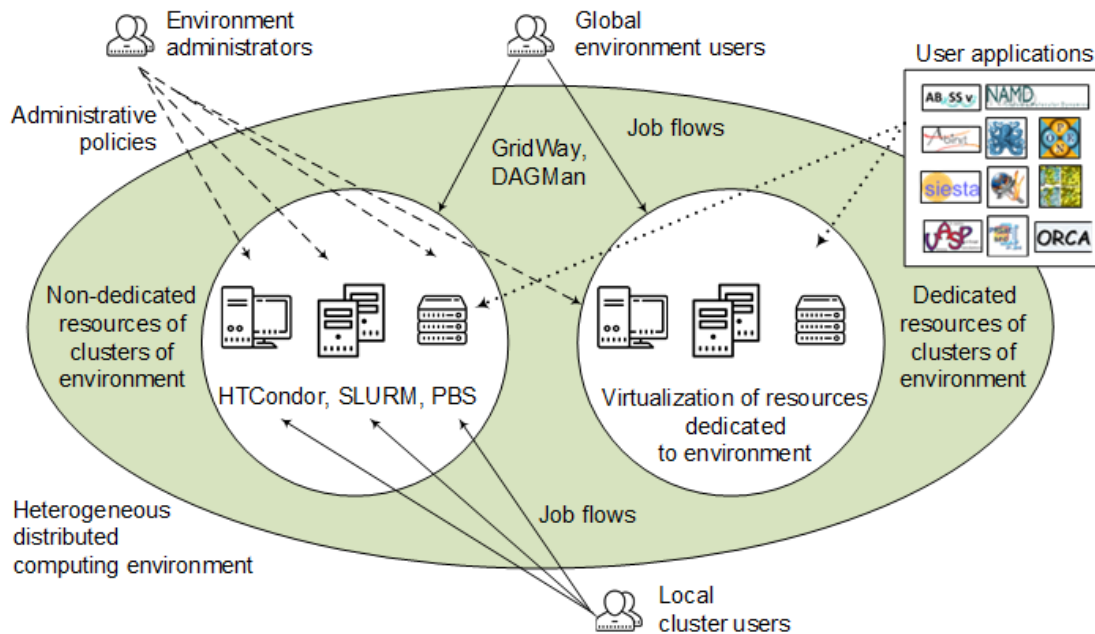


Figure 1. The interaction scheme of the main subjects and objects of the environment.

The heterogeneous distributed computing environment is organized on the basis of the resources of the public access computer center. In this regard, all of its resources are common for users of the center. They are distributed on the basis of administrative policies defined both at the environment level and environment nodes level. The order of the user job execution in the computing environment is strictly regulated.

Administrative policies in computing environment nodes may vary. They establish rules (disciplines) and quotas for resource use by users. In addition, they set the following criteria for the resource use efficiency: loading, balancing, reliability, energy consumption, throughput, fairness of resource allocation, economic parameters (real or virtual), etc.

Depending on the scale of the solved problems, the computing infrastructure of the environment may include personal computers, servers, clusters, grid and cloud platforms. The main components of the environment are computer clusters, including hybrid clusters with heterogeneous nodes. Clusters are organized on the basis of both dedicated and non-dedicated nodes. Therefore, they vary significantly in the degree of reliability of their resources. When the complexity of the modern clusters infrastructure growth, including a growth in the number of their computing elements, the probability of hardware failures and software errors in the problem-solving process significantly increases. Failures occur at random times.

Nodes of different clusters differ in their computational characteristics (the interconnect used, processor performance, number of cores, RAM and disk memory, number of cache levels, etc.), fault tolerance, security, power consumption, computation cost, and other parameters.

The public access computer center allows solving problems for both the local users of clusters and global users who need to use integrated resources of clusters. In the problem-solving process, users of both categories share the common resources of the computing environment.

To solve the problem, the user has to create a job for the environment. The job is a specification of the computational process. It contains information about the required resources, executable application programs, input and output data, and other necessary information. In addition, the job determines the user criteria for solving a problem: time, cost, reliability, safety, and other characteristics.

The computing environment supports the processing of jobs of different classes: sequential and parallel, independent and interconnected, local and remote, parameter sweep (serializable). Jobs also

differ in the time required for its execution, size of the necessary RAM and disk memory, the ratio of real and integer operations, and other computational characteristics. On clusters, local resource managers (LRMs) are installed. LRMs perform job processing. Jobs arrive at the common queues of LRMs. From queues, jobs are sent to execute on the assigned computational resources of the environment. It is assumed that there are not enough free resources in the computing environment to simultaneously service all jobs in queues.

An integration of the cluster environment resources can be performed on the basis of the following technologies:

- Applying LRMs, for example, HTCondor, PBS Torque, or SLURM that support the execution of jobs on several clusters,
- Jobs distribution through the GridWay or Condor DAGMan meta-schedulers,
- Using the Globus Toolkit package as middleware that interacts with LRMs installed on clusters,
- Virtualization of a dedicated part of the environment resources using the OpenStack platform that supports work with a wide spectrum of hypervisors for controlling virtual machines and tools for operating with containers.

In the first case and the second case, the integration is carried out on the basis of the grid-computing model. In the latter case, resource virtualization is performed using the cloud-computing model. Virtualized resources are a private cloud. LRMs are included in virtual entities (virtual machines or containers).

Within the computing environment, there are various applications. They are characterized by varying degrees of the computations scalability, sensitivity to resource heterogeneity, resources virtualization need, and necessity of integrating the model of their subject domain with information about the software and hardware infrastructure of the environment including administrative policies defined for resources. Usually, applications that are sensitive to resource heterogeneity are executed in homogeneous nodes of a cluster or in a virtual environment. If applications can run on heterogeneous resources, their users are interested in maximizing the computing environment performance.

A subject domain model of solved problems is a formalization of the system under study, its objects, their interrelations, and events occurring in it. Usually, a complex system has the following main characteristics:

- A large number of objects,
- Relations of different nature between objects,
- Variety in performed functions of the system,
- Non-trivial system control,
- External stochastic disturbances, etc.

Processes of solving problems in a subject domain are also complex and ambiguous. Therefore, additional information about the environment resources capabilities and acceptable rules of resources use is required for effective solving the problems. Such information allows optimal mapping of computational processes on the structure of the resources used.

In the general case, there is software and hardware redundancy in the computing environment (an application can be placed and executed in different nodes of the environment, and the same computations can be made using various applications). At discrete times, applications generate job flows that compete for the common resources of the environment. In the future, these flows can be joined and re-distributed creating new job flows. Figure 2 schematically shows job flows that are transferred from the user machines to the control nodes and, next, to the clusters. The lines of different types and colors correspond to flows with a different number of jobs.

In a computing environment that integrates virtualized and traditional cluster resources, there is a difficulty in managing job flows owing to the following two main reasons. The first reason is the existing differences in cloud and grid computing. The second reason is conflicts between the

preferences of the resources owners and criteria of the jobs execution quality that are determined by the users of these resources.

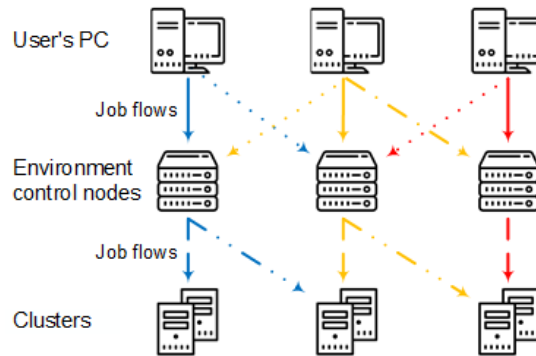


Figure 2. Job flows.

4. Conceptual modeling

We have developed an original conceptual model (figure 3). It provides an interconnected representation of the problem-oriented, software and hardware, simulation, and control layers of knowledge about the integrated computing environment. In addition, the conceptual model allows comprehensively studying the necessary properties (efficiency, reliability, etc.) of scalable scientific and applied applications executed in the environment. It is the basis for a multi-agent system [11] used to manage distributed computing.

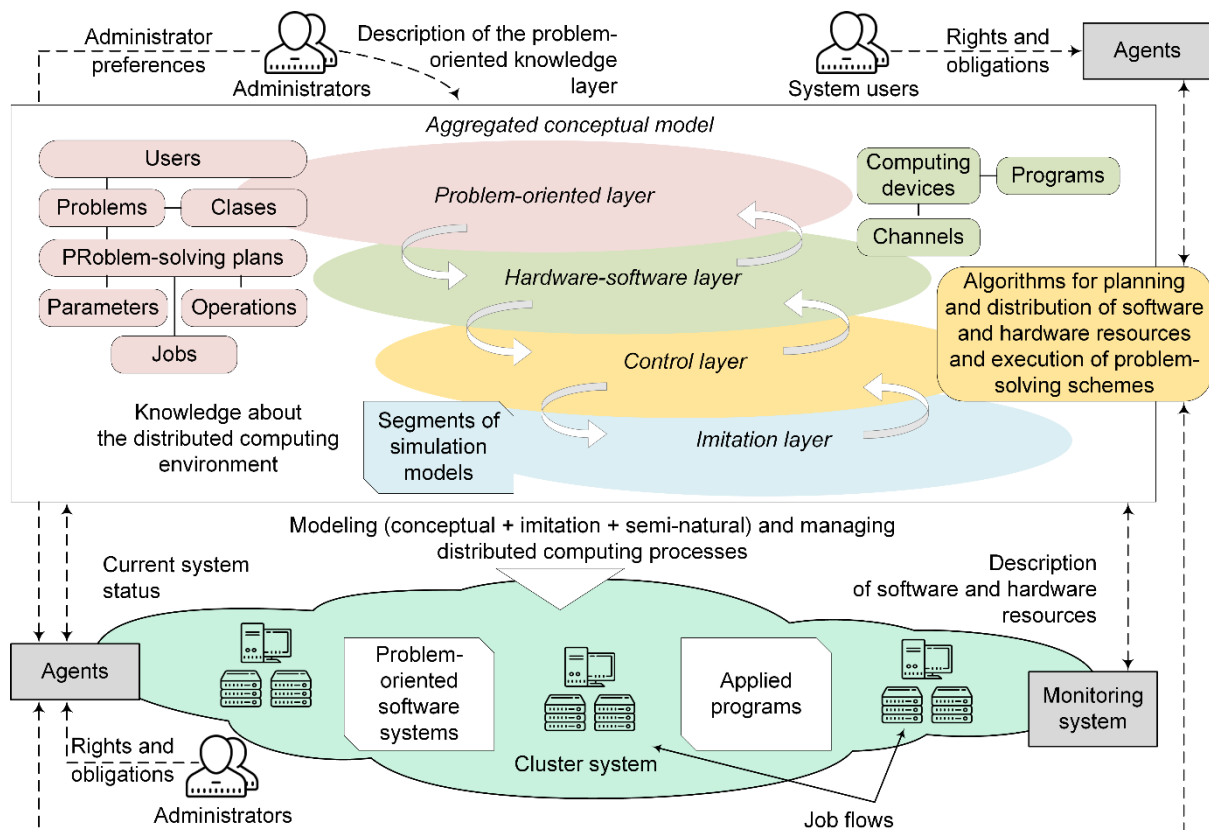


Figure 3. Aggregated conceptual model of the environment.

The model includes the following objects: users, problems, jobs, parameters, operations, nodes, servers, switching devices, data transfer network channels, algorithms, programs, computing processes, and other entities of a subject domain. Different layers of the model contain their own classes of objects. For each object class, its attributes (characteristics) are defined.

The model has an important feature, which in many respects ensures the efficiency of distributed computing management. It provides the ability to identify and account for the specifics of the problems solved in the environment. In particular, the following important characteristics of problems can be highlighted:

- Problem-solving method (using the existed program, designing an algorithm for solving the problem based on the libraries of standard programs, executing the program in the interpretation or compilation modes),
- Location of the program used for solving the problem (in the user's machine or in the environment node),
- Number of the program runs,
- Interrelation of subproblems,
- Type of parallelism for the problem-solving algorithm (fine-grained, large-block, and mixed parallelism),
- Degree of computations intensity (small, medium and large problems for both the computational and data processing),
- Need for the computational process managing in the package or interactive modes,
- Real-time computing and a number of other features of the problem-solving process,
- Information about continuous integration, delivery, and deployment of programs (modules of applications) used in jobs.

Jobs are classified in accordance with the problem characteristics. After defining the classes of jobs, subclasses can be created.

The environment administrator forms the conceptual model [12]. He puts knowledge into the conceptual model in several ways (figure 3). Information about users, problems, jobs, programs, and data come through the administrative subsystem of the environment. The administrator specifies the information of software and hardware resources that is detailed by agents representing environment nodes. He transfers the current state of environment objects through a specialized meta-monitoring system [3].

5. Simulation modeling

Nowadays, simulation modeling is the most effective method for studying distributed computing systems. The SIRIUS III toolkit has been developed at Matrosov Institute for System Dynamics and Control Theory of SB RAS to automate the simulation modeling of heterogeneous distributed computing environments. It supports the automation of parameter sweep computations.

The SIRIUS III toolkit includes the library of standard templates, model designer, library of environment models, executive subsystem, database, and a subsystem of analysis of simulation results (figure 4).

The library of standard templates consists of fragments of the model in the GPSS language. These templates models typical objects of the environment and the processes of their occurrence.

The designer provides the creation of a simulation model of the environment using its conceptual model and the library of standard templates. The created model is included in the environment models library. The executive subsystem provides debugging of environment models and the implementation of parameter sweep computations through these models runs.

To start the model, the corresponding job and sets of variants of the initial data are automatically generated. In addition, the executive subsystem provides for the collection and aggregation of the results of parameter sweep computations. All information obtained from the simulation modeling is

stored in a database. The analysis subsystem is used to study this information based on methods of multi-criteria optimization of model parameters.

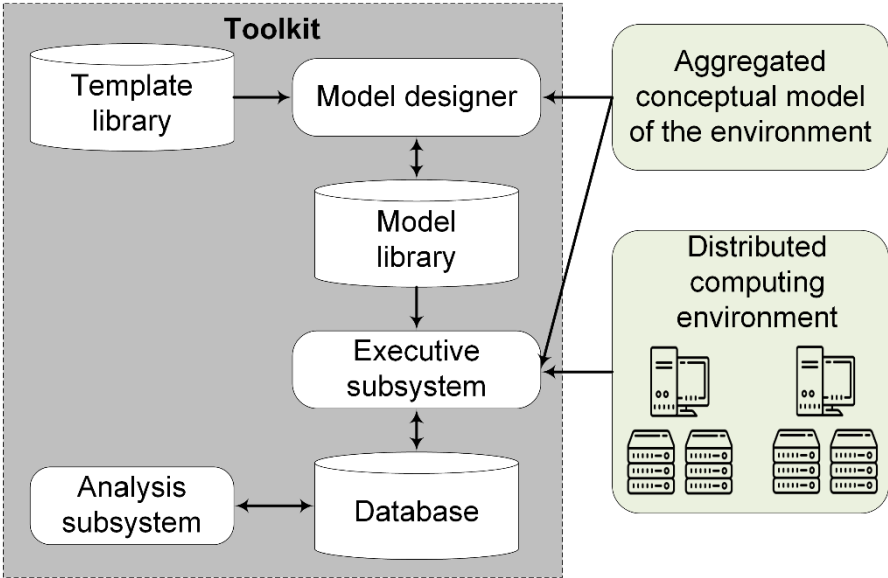


Figure 4. The interaction scheme between the toolkit components.

6. Semi-natural modeling

Semi-natural simulation involves testing the environment by automatically generating and executing synthetic jobs flows in it. The purpose of this modeling is to evaluate the effectiveness of the environment. The synthetic job flow is an artificially generated flow. Simulation programs are used as applications in jobs. They perform the actual computational loading of environment nodes and the exchange of specified amounts of data.

The semi-natural modeling system includes the following components: web-interface, job generation module, scheduler, job manager, and information subsystem (figure 5).

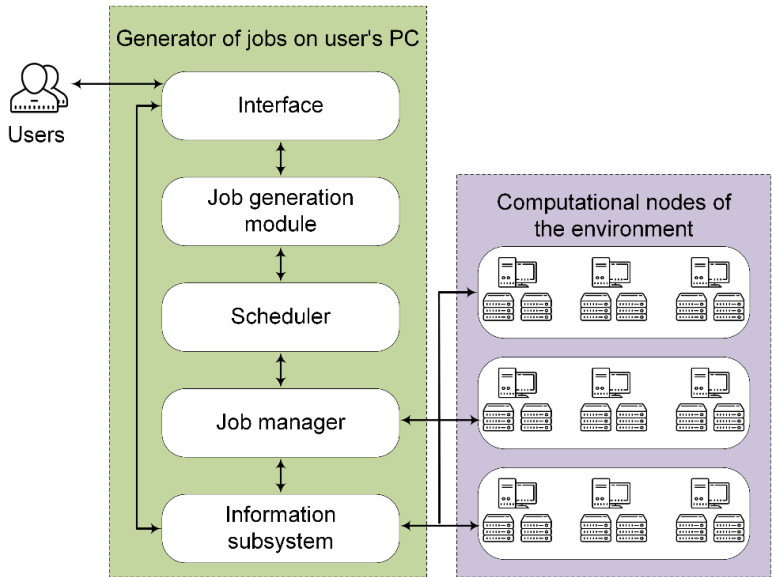


Figure 5. The interaction scheme of components in the semi-natural simulation modeling.

The interface provides the user with the means for specifying test-jobs and obtaining information (statistics) on the progress of jobs. The specification of the test-jobs includes the following characteristics:

- Total number of jobs in the flow,
- Number of jobs of a certain class in the flow,
- Probability distribution laws of the occurrence time for jobs of a certain class,
- Job options (type, execution time, data file sizes, resource requirements, job execution priority, etc.).

Using an intuitive web-interface, the user determines the values of the job flow parameters, its options, and job specifications. The determined values are saved in a text file in the workload format [13].

The job generation module automatically creates special applications (simulators) using user-defined test-job options. Then, it converts test-job specifications from the workload format to the format used by LRM of a computational node. Next, the converted specifications are sent to the scheduler. The scheduler operates in the background and services the test-job queues. In accordance with the specified principle of queue processing and the time for launching a test-job specified in its description, the scheduler submits the job to LRM, which runs it.

The information subsystem is intended for collecting and processing statistical data related to the processes of performing test-jobs. Based on the information received, the environment performance evaluations are made.

7. Technology for testing and evaluating the reliability of the environment

The technology considered in this section represents is the following components:

- Methods and tools for automating the construction of conceptual, imitation and semi-natural environment models,
- Methods and tools for organizing distributed computing and managing them,
- Simulated processes for solving scientific and applied problems,
- Software and hardware environment.

The above-listed components are integrated into a technological chain of testing and evaluating the environment reliability. The automated process of complex modeling of the environment and applying its results includes the following main steps:

- Constructing the aggregated conceptual model of the environment,
- Creating the simulation model of the environment based on the conceptual model,
- Carrying out the experiments through the parameter sweep computations,
- Generalizing and analysis of simulation results,
- Verification of simulation results by carrying out semi-natural modeling,
- Setting the configuration parameters of LRMs in the environment in order to improve its reliability.

Applying the results of testing and evaluating the environment reliability in the distributed computing management [14, 15] can significantly improve the following parameters:

- Number of the unexecuted jobs,
- Number of program restarts,
- Total time to complete a set of interrelated jobs.

In addition, these results enable us to increase the resources efficiency use.

8. Conclusions

In the paper, we propose a new approach to solving problems of testing and evaluating the reliability of heterogeneous distributed computing environments. A feature of the proposed approach is the integration of methods and tools for studying and improving the reliability of problem-oriented

environments based on the conceptual programming paradigms, knowledge bases and meta-monitoring.

We highlight the following main benefits of our approach:

- Applying the aggregated multi-level conceptual models of problem-oriented environments that provide a detailed description of all aspects of problem-solving processes in these environments,
- Combining the analytical, simulation and semi-natural modeling of problem-oriented environments in the process of their study,
- Ensuring the collection, unification, processing, and analysis of relevant information about heterogeneous software and hardware resources of the studied environments on the basis of their meta-monitoring,
- Applying the obtained evaluations to improve the problem-oriented environments functioning reliability through the parameter adjustment of the distributed computing management algorithms used in these environments.

Our further study relies on the fact that information elicited from problem-oriented data is often heterogeneous and subject to frequent changes. To this end, a new model that can represent relations between primary information and data structures used in scientific and applied applications is required.

Acknowledgment. The study is supported by the Russian Foundation of Basic Research, project no. 19-07-00097 (reg. no. AAAA-A19-119062590002-7). This work was also supported in part by the Basic Research Program of SB RAS, project no. IV.38.1.1 (reg. no. AAAA-A17-117032210078-4).

References

- [1] Chen W and Deelman E 2012 Workflowsim: A Toolkit for Simulating Scientific Workflows in Distributed Environments *Proc. 8th Int. Conf. on E-Science* (IEEE) pp 1–8
- [2] Sulistio A, Poduval G, Buyya R and Tham C K 2007 On incorporating differentiated levels of network service into GridSim *Future Gener. Comp. Sy.* **23(4)** 606–615
- [3] 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(2)** 107–112
- [4] Xia H 2004 The MicroGrid: Using Online Simulation to Predict Application Performance in Diverse Grid Network Environments *Proc. of the Workshop on Challenges of Large Applications in Distributed Environments* (IEEE) pp 1-10
- [5] Taura K 2004 Grid Explorer: A Tool for Discovering, Selecting, and Using Distributed Resources Efficiently *IPSIJ SIG Technical Report 2004(81)* (HPC-99) 235-240
- [6] Weins D V, Glinskiy B M and Chernykh I G 2019 Analysis of Means of Simulation Modeling of Parallel Algorithms *Comm. Com. Inf. Sc.* **965** 29–39
- [7] Aida K, Takefusa A, Nakada H, Matsuoka S and Nagashima U 1998 Performance Evaluation Model for Job Scheduling in a Global Computing System *Proc. 7th Int. Symp. on High Performance Distributed Computing* (IEEE) pp 352–353
- [8] Calheiros R, Ranjan R, Beloglazov A, De Rose C Buyya R 2011 CloudSim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms *Software Pract. Exper.* **41(1)** 23-50
- [9] Belalem G and Slimani Y 2007 Consistency Management for Data Grid in Optorsim Simulator *Int. Conf. on Multimedia and Ubiquitous Engineering* (IEEE) pp 554–560
- [10] Access mode: <http://arxiv.org/abs/1309.1630>
- [11] Feoktistov A, Tchernych 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* (Riejka: IEEE) pp 1350–1355
- [12] 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

- [13] Chapin S J, Cirne W, Feitelson D G and al. 1999 Benchmarks and Standards for the Evaluation of Parallel Job Schedulers *Lect. Notes Comput. Sci.* **1659** 66–89
- [14] Feoktistov A, Sidorov I, Tchernykh A, Edelev A, Zorkalzev V, Gorsky S, Kostromin R, Bychkov I and Avetisyan A 2018 Multi-Agent Approach for Dynamic Elasticity of Virtual Machines Provisioning in Heterogeneous Distributed Computing Environment *Proc. of the Int. Conf. on High Performance Computing and Simulation* (IEEE) pp 909–916
- [15] Bychkov I, Feoktistov A, Sidorov I and Kostromin R 2017 Job Flow Management for Virtualized Resources of Heterogeneous Distributed Computing Environment *Procedia Engineering* **201** 534–542