Orlando Tools: Energy Research Application Development Through Convergence of Grid and Cloud Computing

Alexander Feoktistov[1](✉), Sergei Gorsky[1], Ivan Sidorov[1],
Roman Kostromin[1], Alexei Edelev[2], and Lyudmila Massel[2]

[1] Matrosov Institute for System Dynamics and Control Theory of SB RAS,
Irkutsk, Russia
{agf, gorsky, ivan.sidorov, kostromin}@icc.ru
[2] Melentiev Energy Systems Institute of SB RAS, Irkutsk, Russia
{flower, massel}@isem.sei.irk.ru

Abstract. The paper addresses the relevant problem related to the development of scientific applications (applied software packages) to solve large-scale problems in heterogeneous distributed computing environments that can include various infrastructures (clusters, Grid systems, clouds) and provide their integrated use. We propose a new approach to the development of applications for such environments. It is based on the integration of conceptual and modular programming. The application development is implemented with a special framework named Orlando Tools. In comparison to the known tools, used for the development and execution of distributed applications in the current practice, Orlando Tools provides executing application jobs in the integrated environment of virtual machines that include both the dedicated and non-dedicated resources. The distributed computing efficiency is improved through the multi-agent management. Experiments of solving the large-scale practical problems of energy security research show the effectiveness of the developed application for solving the aforementioned problem in the environment that supports the hybrid computational model including Grid and cloud computing.

Keywords: Scientific application · Grid · Cloud · Energy research

1 Introduction

Nowadays, heterogeneous distributed computing environments often integrate resources of public access computing centers using various Grid computing models and cloud infrastructures. Such an integration causes new challenges for computation management systems in a process of solving large-scale problems in the environments. These challenges related to existing differences in the models of cloud and Grid computing, and conflicts between the preferences of resource owners and the quality criteria for solving environment user problems [1].

The multi-agent approach enables to significantly mitigate the above differences and conflicts through interactions of agents representing the resources of centers and clouds, as well as their owners and users. At the same time, coordinating agent actions

V. Voevodin and S. Sobolev (Eds.): RuSCDays 2018, CCIS 965, pp. 289–300, 2019.
https://doi.org/10.1007/978-3-030-05807-4_25

can significantly improve the quality of the computation management, especially in case market mechanisms for regulating supply and demand of resources are used [2].

Another direction for improving the distributed computing quality is the problem-orientation of the management systems [3]. Its importance is due to the need for the effective integrated use of heterogeneous resources of the environment in the process of solving common problems. We should take into account the problem specifics, matching of preferences of resource owners and the problem-solving criteria, and supporting of automatic decision-making in management systems [4, 5].

The effectiveness of the agents functioning directly depends on the knowledge they use [6]. In the known tools for multi-agent computation management [7], processes of elicitation and application of knowledge by agents remain an actual problem and require their development [8].

We propose a new approach to the creation and use of scalable applications that is based on the multi-agent management in the heterogeneous distributed computing environment integrating Grid and cloud computing models. Advantages of the proposed approach are demonstrated by example of an application for solving important large-scale problems of the energy research on the example of energy security field.

Within the proposed approach, the multi-agent system for the computation management is integrated with the toolkits that are used in the public access computer center "Irkutsk Supercomputer Center of the SB RAS" to create scalable applications based on the paradigms of parallel and distributed computing.

The rest of this paper is organized as follows. In the next Section, we give a brief overview of research results related to the convergence of Grid and cloud computing. Section 3 describes the proposed approach to the development and use of scalable applications in the heterogeneous distributed computing environment. Section 4 provides an example of the scalable application for solving the complex practical problem of vulnerability analysis of energy critical infrastructures in terms of energy security including experimental analysis. The last section concludes the paper.

2 Related Work

The scientific application (applied software package) is a complex of programs (modules) intended for solving the certain class of problems in the concrete subject domain. In such application, a computational process is described by problem-solving scheme (workflow). The rapid advancement of technologies for distributed computing has led to significant changes in the architecture of scientific applications [9]. They retained modular structure, but it became distributed.

For a long time, the well-known tools Globus Toolkit [10], HTCondor [11], BOINC [12] or X-COM [13] are used as middleware for distributed application execution. At the same time, workflow management systems are actively developed and applied for the same purpose. Nowadays, there are the systems Askalon, DAGMan Condor, Grid Ant, Grid, Flow, Karajan, Kepler, Pegasus, Taverna, Triana, etc. [14]. Often, workflow management systems and middleware are used together.

However, the possibilities of the above tools and systems often restrict potential opportunities of the modern scientific applications within the certain computational model [15]. In this regard, the research field related to the integration of various computational models becomes very topical.

Integration of the cloud and Grid computing models leads to the need to solve problems interface unification, application adaptation and their scalability, providing quality of service, mitigating uncertainty of different kinds, intellectualization of resource management system, monitoring heterogeneous resources, etc. A lot of scientific and practical results were provided to solve above problems the last ten years. We brief overview some of them.

Rings et al. [16] propose an approach to the convergence of the Grid and cloud technologies within the next generation network through the design of standards for interfaces and environments that supports multiple realizations of architectural components.

Kim et al. [17] study efficiency of the hybrid platform when changing resource requirements, quality of service and application adaptability.

Mateescu et al. [18] represent a hybrid architecture of high-performance computing infrastructure that provides predictable execution of scientific applications and scalability when a number of resources with different characteristics, owners, policies and geographic locations are changed.

A cloud application platform Aneka is implemented to provide resources of the various infrastructures, including clouds, clusters, Grids, and desktop Grids [19].

An example of the web-oriented platform that supports the use of different computational systems (PC, computational cluster, cloud) is represented in [20]. The subject domain specific of this platform is computational analyses of genomic data.

The Globus Genomics project provides cloud-hosted software service for the rapid analysis of biomedical data [21]. It enables to automate analysis of large genetic sequence datasets and hide the details of the Grid or cloud computing implementation.

Mariotti et al. [22] propose an approach to the integration of Grids and cloud resources using data base management system for deploying virtual machine (VM) images in cloud environments on requests from Grid applications.

Talia [6] analyses cloud computing and multi-agent systems. He shows that many improvements in distributed computing effectiveness can be obtained on the base of their integrated use. Among them, provisioning powerful, reliable, predictable and scalable computing infrastructure for multi-agent based applications, as well as making Cloud computing systems more adaptive, flexible, and autonomic for resource management, service provisioning and executing large-scale applications.

To this end, we provide a special framework named Orlando Tools for developing scalable applications and creating the heterogeneous distributed computing environment that can integrate Grid and cloud computing models. In addition, we apply a multi-agent system to improve the job flow management for the developed applications. Thus, in comparison with aforementioned projects, we ensure developing and using the joint computational environment included virtualized Grid and cloud resources under the multi-agent management.

3 Orlando Tools

Orlando Tools is the special framework for the development and use of scientific applications (applied software packages) in heterogeneous distributed computing environments. It includes the following main components:

- Web-interface supporting user access to other components of Orlando Tools,
- Model designer that is applied to specify knowledge about an application subject domain in both the text and graphic modes (in text mode, a knowledge specification is described in the XML terms),
- Knowledge base with information about the application modular structure (sets of applied and system modules), schemes of a subject domain study (parameters, operations and productions of a subject domain, and their relations), hardware and software infrastructure (characteristics of the nodes, communication channels, network devices, network topology, failures of software and hardware, etc.),
- Executive subsystem providing the problem-solving scheme schedulers and scheme interpreters that use the subject domain specification (computational model) for the distributed computing management at the application level,
- Computation database, which stores parameter values used in the problem-solving processes.

Orlando Tools provides an integration of the developed applications. The model designer enables application developers to use fragments of subject domain descriptions, software modules, input data and computation results of other applications in the process of creating a new application. Therefore, the time needed to develop applications and carry out experiments is reduced. Figure 1 shows an integration scheme of computational infrastructures into the joint environment with Orlando Tools.

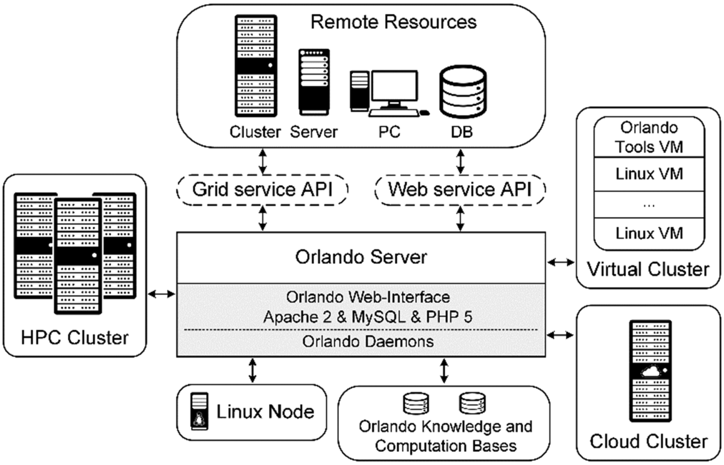

Fig. 1. Integration scheme of computational infrastructures into a heterogeneous distributed computing environment with Orlando Tools.

The integration of computational infrastructures is carried out through Orlando Server. It provides the Web-Interface and Daemons that implement functions of the executive subsystem in the automatic mode. Orlando Server is placed in the dedicated or non-dedicated computational nodes. The Server enables to include the following infrastructures into the integrated environment:

- HPC-clusters with the local resource manager system PBS Torque,
- Linux nodes (individual nodes with operating system Linux) that can be used to include non-portable (located in specialized nodes) software in an application,
- Virtual clusters that are created by using non-cloud resources with the special VMs of Orlando Tools in the images of which is placed PBS Torque (Fig. 2),
- Cloud clusters that are created using cloud resources with the Orlando Tools VMs,
- Remote resources that are included through the Grid or Web service API.

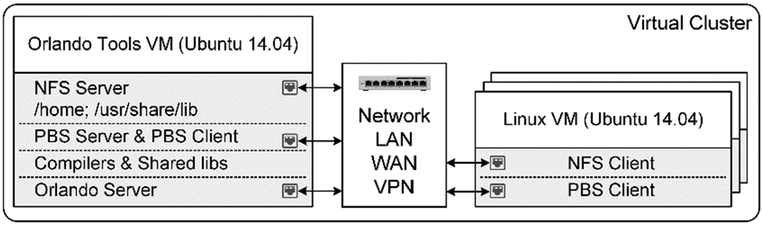

Fig. 2. Architecture of the virtual cluster

The images Orlando Tools VM and Linux VM are preconfigured and packed using Open Virtualization Format – Version 1.0 that provides the VM portability on different hardware and compatibility with various hypervisors.

Nodes have one of the hypervisors (Oracle VM VirtualBox, ESXi, XEN). The virtual cluster is created through placing VMs in the nodes united by LAN, WAN or VPN. NFS Server is used to provide shared access of Linux VM to DLL (/usr/share/lib), application programs and data (/home) on the network. An access to Orlando Tools is carried out by IP-address of Orlando Tools VM.

Applications that are developed in Orlando Tools generate job flows. The job describes the problem-solving process and includes information about the required computational resources, used programs, input and output data, communication network and other necessary data. Job flows are transferred to computational infrastructures that are included in the environment. The Orlando Tools scheduler decomposes job flows between the infrastructures taking into account the performance of their nodes relative to the problem-solving scheme. The node performance evaluation is obtained through the preliminary experiments with application modules. When job flows are transferred to the environment based on the resources of the Irkutsk Supercomputer Center, they are managed by multi-agent system. In this case, the distributed computing efficiency can be significantly improved [23]. The Orlando Tools architecture and model of application subject domains are considered in detail in [24].

4 Energy Research Application

4.1 Problem Formulation

One of the main energy security (ES) aspects is the ensuring conditions for the maximum satisfaction of consumers with energy resources in emergency situations. Investigation of this ES aspect requires the identification of critically important objects (CIO) in the energy sector in general or a particular energy system. CIO is a facility, partial or complete failure of which causes significant damage to the economy and society from the energy sector side.

Today, more than 90% of Russian natural gas is extracted in the Nadym-Pur-Tazovsky district of the Tyumen region. The distance between that district and the main natural gas consumption areas of Russia is 2–2.5 thousand km and the countries that import Russian natural gas are located 2–3 thousand km further. Thus, practically all Russian natural gas is transmitted for long distances through the system of pipelines. It has a number of mutual intersections and bridges, moreover, the pipes of essential gas pipelines are often laid near each other. Some intersections of main pipelines are extremely vital for the normal natural gas supply system operation.

The technique of identifying critical elements in technical infrastructure networks [25] is used to determine CIO of the gas supply system of Russia [26]. The criticality of an element or a set of elements is defined as the vulnerability of the system to failure in a specific element, or set of elements. An element is critical in itself if its failure causes severe consequences for the system as a whole.

Identifying critical elements is usually a simple task when only dealing with single failures. It becomes difficult when considering multiple simultaneous failures. A single element failure or multiple simultaneous element failures are referred to in [25] as failure sets. A failure set is a specific combination of failed elements and is characterized by a size, which is equal the number of elements that fail simultaneously. The investigation of failure sets with synergistic consequences is especially difficult because those consequences cannot be calculated by summarizing the consequences of the individual failures. For example, synergistic consequences of failure sets of size 3 cannot be determined by summarizing the consequences of failure subsets of size 2 and size 1. The number k of possible combinations of failed elements to investigate is defined by the following formula:

$$k = \frac{m!}{(m-n)!n!}, \tag{1}$$

where m is the number of system elements to fail and n is the size of the failure sets.

In formula (1), the number k increases rapidly in case n become greater. Evaluating all possible combinations of failures is practically impossible for a personal computer. Reducing k leads to losing important information about the system's vulnerability to failures. For example, the considering of only combinations of elements that are critical by themselves can miss "hidden" elements. The single failure of these elements causes little consequences for the system. At the same time, combining them with other elements results the failure sets with large synergistic consequences.

Another way to get rid of the rapid growth of k in the real energy systems with big m is applying high-performance computing for the analysis of system's vulnerability to failure sets (also without synergistic effects).

4.2 Application Subject Domain Structure

The process of organization of scientific research groups evolves towards virtual geographically distributed groups working on a project. It is necessary to ensure the availability of information and computing resources of the project for all its participants [27]. An energy research environment consists of computational, information and telecommunication infrastructures. The concept of creating an energy research environment is methodologically justified using the fractal stratified model (FS-model) of information space [28]. The FS-model allows mapping all available domain information into a set of interrelated layers that unite information objects, which have the same set of properties or characteristics. Each layer in turn can be stratified. FS-modeling represents an IT-technology as a set of information layers and their mappings. The IT-technology includes tools to describe information layers and facilities to support mappings from any layer to each.

Graphically, the FS-model of an energy research environment is represented as a set of nested spherical layers defined by the triple $\{S, F, G\}$ (Fig. 3), where S denotes set of layers, F is the set of mappings, and G is the set of invariants.

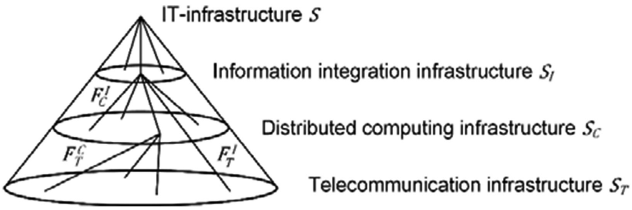
IT-infrastructure S
Information integration infrastructure S_I
Distributed computing infrastructure S_C
Telecommunication infrastructure S_T
F_C^I
F_T^C
F_T^I

Fig. 3. The FS-model of an energy research environment

An energy research environment S according to the FS-methodology is stratified into the information integration infrastructure S_I, distributed computing infrastructure S_C and telecommunications infrastructure S_T. S_I is layered into data and the metadata layers (S_{ID}, S_{IM}), S_C is layered into programs and their meta description layers (S_{CP}, S_{CM}). In the set F, there are the following mappings: $F_C^I : S_I \to S_C$, $F_T^I : S_I \to S_T$, $F_T^C : S_C \to S_T$. The invariants G denote energy research objectives detailed for each layer. The information models, data structure models, and ontologies are used to describe the meta-layers.

Orlando Tools is used for the mapping support when the energy research environment is created. Schemes of its knowledge and databases relate to S_I. The module structure reflects S_C. Information about the hardware and software infrastructure ensures to implement S_T. Relations between objects of the computational model provide F_C^I, F_T^I and F_T^C. Depending of the energy research type, for example, the FS-model for the vulnerability analysis of energy critical infrastructures [26] or the energy sector development investigation [24] can be described. Merging both of them into one constitutes the FS-model of ES research environment. Thus, the researcher creating different FS-models can build the variety of energy research environments on the basis of Orlando Tools using the same set of modules.

4.3 Computational Experiment

The modern gas supply system of Russia model consists of 378 nodes, including: 28 natural gas sources, 64 consumers (Russian Federation regions), 24 underground gas storages, 266 main compressor stations, and 486 arcs which represent main transmission pipelines segments and outlets to distribution networks.

In the first experiment 415 arcs were chosen as elements to fail and the failure sets of size 2 and 3 were considered. The experiment results are represented in Table 1, where v shows the total natural gas shortage in percentages, $h_{1,n=2}$ and $h_{2,n=3}$ express the contribution of a specific element's synergistic consequences to the total synergistic consequences for size n failure sets. According to the second column of Table 1, the single failure of elements from A to E leads to the significant gas shortage from 15 to 21% of the total system demand. Thus, they can be identified as CIO of the gas supply system of Russia. In opposite, elements from N to T are not CIO because the total natural gas shortage due to their failure is less than 5%.

Table 1. The criticality of main natural gas pipelines segments

Element	v, %	$h_{1,n=2}$	$h_{2,n=3}$	Element	v, %	$h_{1,n=2}$	$h_{2,n=3}$
A	21	0.0739	0.0756	K	8	0.0313	0.0386
B	21	0.0752	0.0784	L	6	0.0218	0.0263
C	21	0.0763	0.0807	M	5	0.0237	0.0318
D	15	0.0573	0.0614	N	4	0.0255	0.0348
E	15	0.0573	0.0614	O	4	0.0303	0.0411
F	9	0.0352	0.0424	P	4	0.0302	0.0410
G	9	0.0351	0.0423	Q	4	0.0161	0.0211
H	8	0.0327	0.0401	R	4	0.0161	0.0211
I	8	0.0354	0.0418	S	3	0.0226	0.0325
J	8	0.0369	0.0444	T	3	0.0227	0.0324

The measures $h_{1,n=2}$ and $h_{2,n=3}$ in Table 1 are used to prioritize preemptive efforts to reduce system-wide vulnerability. An element with large synergistic consequences would score higher on the preparedness activity than the other ones with the same total natural gas shortage value. For example, the element C is more preferable than A and B for the implementation of preparedness options. The obtained experimental results allowed forming the new recommendations to correct the importance of CIO that affects the budget of preparedness options and order of their implementation.

The experiment was performed in a heterogeneous distributed computing environment that is created through applying Orlando Tools and based on the resources of the Irkutsk Supercomputer Center [29]. We used the following node pools:

(1) 10 nodes with 2 processors Intel Xeon 5345 EM64T (4 core, 2.3 GHz, 8 GB of RAM) for each (non-dedicated resources),
(2) 10 nodes with 2 processors AMD Opteron 6276 (16 core, 2.3 GHz, 64 GB of RAM) for each (dedicated resources),
(3) 10 nodes with 2 processors Intel Xeon CPU X5670 (18 core, 2.1 GHz, 128 GB of RAM) for each (dedicated and non-dedicated resources).

Pool nodes have also various types of the interconnection (1 GigE, QDR Infiniband) and hard disks (HDD, SSD). All dedicated resources are virtualized.

4.4 Experimental Analysis

Table 2 shows the problem-solving time for $n \in \{2, 3, 4\}$. We demonstrate the following parameters: number l of possible failure sets, time $t_{k=1}$ of solving the problem on PC with the processor Intel Core i5-3450 (4 core, 3.10 GHz, 4 GB of RAM), time $t_{k=400}$ of solving the problem in the pools 1 and 2, and time $t_{k=760}$ of solving the problem in the pools 1–3, where k is the maximum number of available cores.

Table 2. Problem-solving time

n	l	$t_{k=1}$	$t_{k=400}$	$t_{k=760}$
2	85905	16	67	70
3	11826255	656640	7200	2730
4	1218104265	31536000	1555200	622080

We obtained the $t_{k=1}$, $t_{k=400}$ and $t_{k=760}$ values for $n = 2$ and $n = 3$ through the real experiments with Orlando Tools and then evaluated the $t_{k=1}$, $t_{k=400}$ and $t_{k=760}$ values for $n = 4$. It is obvious that the augment of n affects to the rapid rise of l thereby increasing the problem-solving time in many times.

The expediency of computing on PC is shown only for $n = 2$. In this case, the problem-solving time in the pools is the time obtained on PC due to the existence of overheads associated with the transfer of data between pool nodes. For $n = 3$, the problem-solving time in the pools is significantly lower the time shown on PC. When $n = 4$, problem-solving on PC is practically impossible.

Figure 4a and b show a number of the used cores and slots for solving the problems with $n = 3$ and $k = 400$ or $k = 760$. A job flow decomposition between the pools taking into account the performance of their nodes ensures the problem-solving time decrease about 6%.

In additional, we apply the special hypervisor shell for multi-agent management of jobs when dedicated and non-dedicated resources were used [25]. The management improvement is achieved by using the problem-oriented knowledge and information about software and hardware of the environment that are elicited in the process of the job classification, and parameter adjustment of agents. The hypervisor shell ensures to launch VMs in both the dedicated and non-dedicated resources within the framework of the joint virtualized environment. It also allows to use free slots in schedules of local resource manager systems of non-dedicated resources. We evaluate the problem-solving time decrease over 9% due to the hypervisor shell applying. Thus, the job flows decomposition and hypervisor shell use provide the significant problem-solving time decrease over 15%.

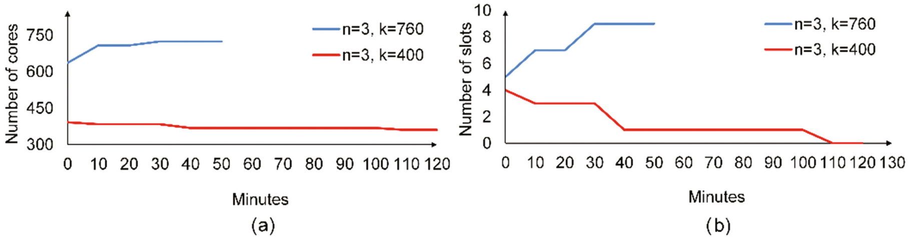

Fig. 4. Number of the used cores (a) and slots (b)

Figure 5a demonstrates the high average CPU load. It is a bit less in the pools 1 and 3 due to overheads of virtualization. At the same time, the virtualization overheads in nodes of all pools were less than 5%. Figure 5b shows the improvement (decrease) in the problem-solving time with $n = 3$ in both $k = 400$ and $k = 760$.

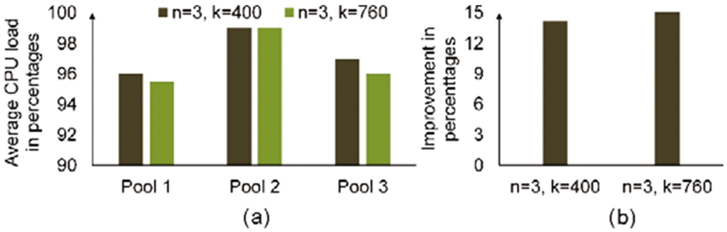

Fig. 5. Average CPU load in pools (a) and the improvement of problem-solving time (b)

5 Conclusions

Nowadays, Grid technologies continue to play important role in the development of scientific computing environments. At the same time, clouds are quickly evolving. The area of their applying constantly expands and often intersects with the field of Grid computing. In this regard, there is a want to use benefits both Grid and cloud computing.

To this end, we propose a new approach to the development and use of scalable applications, and create the special framework named Orlando Tools to support it.

Orlando Tools provides the opportunity to not only develop applications, but also include various computational infrastructures (individual nodes, clusters, Grids and clouds) in the heterogeneous distributed computing environment and share their possibilities and advantages in the problem-solving process. Thus, the capabilities of developed applications can be supported by the needed part of the environment.

We demonstrate the benefits of the proposed approach by example of an application for solving important large-scale problems of research on energy security. A problem solution enables to clarify the previous study results. The experimental analysis shows the advantages and drawback in the use of different infrastructures, and improvement of distributed computing efficiency under the multi-agent management.

Acknowledgements. The work was partially supported by Russian Foundation for Basic Research (RFBR), projects no. 16-07-00931, and Presidium RAS, program no. 27, project “Methods and tools for solving hard-search problems with supercomputers”. Part of the work was supported by the basic research program of the SB RAS, project no. III.17.5.1.

References

1. Toporkov, V., Yemelyanov, D., Toporkova, A.: Anticipation scheduling in grid with stakeholders preferences. Commun. Comput. Inf. Sci. **793**, 482–493 (2017)
2. Sokolinsky, L.B., Shamakina, A.V.: Methods of resource management in problem-oriented computing environment. Program. Comput. Soft. **42**(1), 17–26 (2016)
3. Singh, A., Juneja, D., Malhotra, M.: A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. J. King Saud Univ. Comput. Inf. Sci. **29**(1), 19–28 (2015)
4. Singh, S., Chana, I.: QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput. Surv. **48**(3) (2016). Article no. 42
5. Qin, H., Zhu, L.: Subject oriented autonomic cloud data center networks model. J. Data Anal. Inform. Process. **5**(3), 87–95 (2017)
6. Talia, D.: Cloud computing and software agents: towards cloud intelligent services. In: Proceedings of the 12th Workshop on Objects and Agents. CEUR Workshop Proceedings, vol. 741, pp. 2–6 (2011)
7. Madni, S.H.H., Latiff, M.S.A., Coulibaly, Y.: Recent advancements in resource allocation techniques for cloud computing environment: a systematic review. Cluster Comput. **20**(3), 2489–2533 (2017)
8. Farahani, A., Nazemi, E., Cabri, G., Capodieci, N.: Enabling autonomic computing support for the JADE agent platform. Scalable Comput. Pract. Exp. **18**(1), 91–103 (2017)
9. Ilin, V.P., Skopin, I.N.: Computational programming technologies. Program. Comput. Soft. **37**(4), 210–222 (2011)
10. Foster, I., Kesselman, C.: Globus: a metacomputing infrastructure toolkit. Int. J. High Perform. Comput. Appl. **11**(2), 115–128 (1997)
11. Couvares, P., Kosar, T., Roy, A., Weber, J., Wenger, K.: Workflow management in condor. In: Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M. (eds.) Workflows for e-Science, pp. 357–375. Springer, London (2007). https://doi.org/10.1007/978-1-84628-757-2_22
12. Anderson, D.: Boinc: a system for public-resource computing and storage. In: Buyya, R. (ed.) Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, pp. 4–10. IEEE (2004)

13. Voevodin, V.V.: The solution of large problems in distributed computational media. Automat. Remote Control **68**(5), 773–786 (2007)
14. Talia, D.: Workflow systems for science: concepts and tools. ISRN Soft. Eng. **2013** (2013). Article ID 404525
15. Tao, J., Kolodziej, J., Ranjan, R., Jayaraman, P., Buyya, R.: A note on new trends in data-aware scheduling and resource provisioning in modern HPC systems. Future Gener. Comput. Syst. **51**(C), 45–46 (2015)
16. Rings, T., et al.: Grid and cloud computing: opportunities for integration with the next generation network. J. Grid Comput. **7**(3) (2009). Article no. 375
17. Kim, H., el-Khamra, Y., Jha, S., Parashar, M.: Exploring application and infrastructure adaptation on hybrid grid-cloud infrastructure. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, pp. 402–412. ACM (2010)
18. Mateescu, G., Gentzsch, W., Ribbens, C.J.: Hybrid computing – where HPC meets grid and cloud computing. Future Gener. Comput. Syst. **27**(5), 440–453 (2011)
19. Vecchiola, C., Calheiros, R.N., Karunamoorthy, D., Buyya, R.: Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka. Future Gener. Comput. Syst. **28**(1), 58–65 (2012)
20. Goecks, J., Nekrutenko, A., Taylor, J.: Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. **11**(8), 1–13 (2010)
21. Allen, B., et al.: Globus: a case study in software as a service for scientists. In: Proceedings of the 8th Workshop on Scientific Cloud Computing, pp. 25–32. ACM (2017)
22. Mariotti, M., Gervasi, O., Vella, F., Cuzzocrea, A., Costantini, A.: Strategies and systems towards grids and clouds integration: a DBMS-based solution. Future Gener. Comput. Syst. (2017). https://doi.org/10.1016/j.future.2017.02.047
23. Feoktistov, A., Sidorov, I., Sergeev, V., Kostromin, R., Bogdanova, V.: Virtualization of heterogeneous HPC-clusters based on OpenStack platform. Bull. S. Ural State Univ. Ser. Comput. Math. Soft. Eng. **6**(2), 37–48 (2017)
24. Edelev, A., Zorkaltsev, V., Gorsky, S., Doan, V.B., Nguyen, H.N.: The combinatorial modelling approach to study sustainable energy development of Vietnam. Commun. Comput. Inf. Sci. **793**, 207–218 (2017)
25. Jonsson, H., Johansson, J., Johansson, H.: Identifying critical components in technical infrastructure networks. Proc. Inst. Mech. Eng. O J. Risk Reliab. **222**(2), 235–243 (2008)
26. Vorobev, S., Edelev, A.: Analysis of the importance of critical objects of the gas industry with the method of determining critical elements in networks of technical infrastructures. In: 10th International Conference on Management of Large-Scale System Development (MLSD). IEEE (2017). https://doi.org/10.1109/mlsd.2017.8109707
27. Massel, L.V., Kopaygorodsky, A.N., Chernousov, A.V.: IT-infrastructure of research activities realized for the power engineering system researches. In: 10th International Conference on Computer Science and Information Technologies, pp. 106–111. Ufa State Aviation Technical University (2008)
28. Massel, L.V., Arshinsky, V.L., Massel, A.G.: Intelligent computing on the basis of cognitive and event modeling and its application in energy security studies. Int. J. Energy Optim. Eng. **3**(1), 83–91 (2014)
29. Irkutsk Supercomputer Center of SB RAS. http://hpc.icc.ru. Accessed 13 Apr 2018