Multi-Agent Approach for Dynamic Elasticity of Virtual Machines Provisioning in Heterogeneous Distributed Computing Environment

Alexander Feoktistov[1], Ivan Sidorov[2]
Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia
[1]agf65@icc.ru, [2]ivan.sidorov@icc.ru

Andrei Tchernykh[3,*]
CICESE Research Center, Ensenada, Mexico
Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia
South Ural State University, Chelyabinsk, Russia
[3]chernykh@cicese.mx

Alexei Edelev[4], Valery Zorkalzev[5]
Melentiev Energy Systems Institute of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia
[4]flower@isem.sei.irk.ru, [5]zork@isem.sei.irk.ru

Roman Kostromin[6], Sergey Gorsky[7], Igor Bychkov[8]
Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia
[6]kostromin@icc.ru, [7]gorsky@icc.ru, [8]bychkov@icc.ru

Arutyun Avetisyan[9]
Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia
[9]arut@ispras.ru

Abstract **– We address dynamic elasticity issues of VM provisioning in a heterogeneous distributed computing environment that integrates resources of a data center. We consider the scenario when the center includes dedicated resources (private cloud) for providing virtualized service and non-dedicated resources for Grid-computing. Existing platforms for the resource virtualization does not support management of such an infrastructure. We propose an approach to a job management based on the dynamic elasticity of virtual machines provisioning using resources of both types. We develop the multi-agent job scheduler for dedicated resources and hypervisor shell to launch virtual machines through queues of resource management systems in non-dedicated resources. The scheduler provides a dynamic elasticity of virtual machine provisioning. Advantages of the offered approach to the resource virtualization are demonstrated by an example of a job flow management for a scalable application to solve the complicated practical problem. It is related to the energy security of Vietnam. Provided experiments show that using the developed tools together with the platform for the resource virtualization enables agents to significantly speed up the problem-solving process.**

Keywords – cloud; Grid; virtual machines; multi-agent management

I. Introduction

An effective solving large scientific and applied problems requires the use of high-performance computing systems. However, they are significantly different in computational capabilities, hardware and software platforms, and system architectures.

Their integration into a joint heterogeneous distributed computing environment and virtualization of their resources within the private cloud provides the necessary flexibility for supporting large-scale and complicated-structure computations.

Virtual machines (VMs) are managed by hypervisors. However, the known hypervisors and middleware do not support interaction with external resource management systems (RMS) used in non-dedicated resources. This fact prevents effective, coordinated management of a computational load in dedicated and non-dedicated resources.

The computational load is determined by a set of user application jobs. The job is described by a problem-solving process specification. It includes information about the required computational resources, used software, input and output data, communication network, and other necessary data.

Often, management systems have incomplete information about workflow, and uncertainty of job submission and execution time in resources that have different computational characteristics. Therefore, there is a problem of the dynamic elasticity of VM provisioning taking into account the matching of job properties and resource characteristics.

Dynamic partitioning and coordinated use of dedicated and non-dedicated resources in a problem-solving process are important to increase efficiency and cost reduction.

* Corresponding author

978-1-5386-7879-4/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCS.2018.00145

In the paper, we describe a new approach to the integration and coordinated use of cluster resources of the public access computer center “Irkutsk supercomputer center of Siberian Branch of the Russian Academy of Sciences” [1]. Our objective is to provide users with dynamic expanding available private cloud resources through the dynamic elasticity of VM provisioning in non-dedicated resources. We support it taking into account restrictions on the resource use accepted at the computer center.

We create the joint environment with dedicated and virtualized resources for executing jobs of scalable applications under the OpenStack platform management [2]. This environment refers to the private cloud. A cloud infrastructure is provisioned for exclusive use by registered users of the center. It is owned, managed, and operated by the center. We develop additional specialized tools and use them together with the OpenStack platform. Users are provided with a uniform VM environment that operates both in dedicated and non-dedicated resources.

We present the parameter sweep application that implements the brute force method for a search of the optimal solution of a large-scale problem. It is related to the energy development of Vietnam taking into account the energy security requirements. The application shows the advantages of the proposed approach in comparison with traditional resource virtualization techniques.

The rest of the paper is organized as follows. In the next Section, we give a brief overview of approaches related to the integration and virtualization of computational resources. Section 3 describes the main aspects of the proposed approach to a job execution based on the dynamic elasticity of VM provisioning in the heterogeneous distributed computing environment. Section 4 describes an example of a job flow management for a scalable application to solve the complex practical problem. Section 5 concludes with a summary of our study.

II. RELATED WORK

The problems of dynamic integration of traditional and virtualized computer resources into the unified environment to support large-scale scientific workflows and elasticity are not adequately addressed in the scientific literature.

An analysis of state of the art in this field shows that there is no common approach to solve this problem.

Herres-Pawlis et al. [3] provide the specialized package to integrate Grid systems for various purposes.

Megino et al. [4] describe new features of the Production and Distributed Analysis (PanDA) system used for the workload management within the large-scale ATLAS (A Toroidal LHC ApparatuS) experiment at the Large Hadron Collider and other scientific problems.

Allen et al. [5] propose the elastic cloud platform Globus that support implementation, deployment, and operation of SaaS services for virtual machine environments.

Manvi et al. [6] consider some of the important resource management techniques such as resource provisioning, resource allocation, resource mapping, resource adaptation, and dynamical workflow scaling to resources.

It is important to note that the dynamic resource integration can significantly improve a problem-solving process. When we know a problem subject domain, we can allocate needed resources for their execution to scale resources if it is necessary and select the optimal problem-solving algorithms.

Sokolinsky and Shamakina [7] propose a new scheduling algorithm that unlike known ones takes into account specification of the problem-oriented jobs and provides a greater degree of computation efficiency.

Inggs et al. [8] introduce a domain specifics approach for modeling job run-time characteristics in heterogeneous computing platforms. The authors evaluate the efficiency of different resource allocation algorithms.

Kliazovich et al. [9] study performance issues of resource allocation in cloud computing and propose a Communication-Aware Directed Acyclic Graphs (CA-DAG) model of applications that allows making separate resource allocation decisions: assigning processors to handle computing jobs, and network resources for information transmissions.

Tchernykh et al. [10] discuss techniques for mitigating uncertainty and improving efficiency of resource provisioning. Ramírez-Velarde et al. [11] study adaptive resource allocation under uncertainty.

Multi-Agent Systems (MASs) provide the preference coordination for owners and users of resources. For example, Singh [12] presents an agent framework for supporting scalability in cloud computing environments by finding a cloud that fully satisfied user requests when the used cloud is overloaded.

Kalyaev et al. [13] present the multi-agent scheduling in cloud computing environments taking into account job completion deadlines.

Moreover, the use of market mechanisms for the regulation of resources supply and demand significantly improves resource allocation for job execution.

Toporkov and Yemelyanov [14] show that the economic model of scheduling and fair resource sharing in distributed computations significantly improve job management quality and resource usage taking into account specific features of management policy of resource allocation and consumption, as well as job properties and problem-solving criteria.

Sharma et al. [15] demonstrate advantages of a risk-adjusted cloud resources pricing model that uses concepts and algorithms of financial option theory for providing the agreeing on the preferences of cloud providers and users.

However, known MASs often do not provide interactions with other systems used for supporting interconnected jobs, carrying out parameter sweep applications, management of cluster and Grid-resources. In current MAS systems, the use of job subject domain for scheduling optimization is limited.

Two approaches to the resource virtualization are now actively developed: virtual machine-based and container-based.

The main idea of the first approach consists in emulating hardware components with the required characteristics within the VM with the guest operating system. Various tools are used for VM-based virtualization: VMware, Kernel-based Virtual Machine (KVM), and Xen [16].

The container-based approach allows running multiple isolated user-space instances on the same kernel. Linux containers (LXC) and Docker are examples of the container-based virtualization systems [17].

The known hypervisors and container management systems include the following characteristics:

- Overheads for emulating hardware,
- Changing resource properties,
- Launch speed of a virtual object,
- Supporting guest operating systems and own file system,
- Creating virtual objects in heterogeneous environments and providing their isolation from neighbors,
- Security degree,
- Supporting virtual private network,
- Managing sockets and processes,
- Providing free software,
- Interaction with external RMSs, etc.

The comparative analysis of these characteristics shows that all of them have certain advantages and drawbacks [18, 19]. We see that VMs are more expedient for the organization of scalable computations in the heterogeneous distributed computing environment with different users and a variety of job classes. The effective solving of large-scale problems often requires applying middleware for managing of different virtual elements with the hypervisors mentioned above and container management systems.

The platforms with open source code as OpenStack, Apache CloudStack [20], Eucalyptus [21], and Open Nebula [22] can be used as such middleware. The OpenStack platform is the indisputable leader. However, the careful configuration of its components is required for the efficient functioning of the platform.

It should be noted that the tools mentioned above do not interact with external RMSs. Therefore, it is difficult to ensure the effective VM provisioning in non-dedicated resources.

III. Job Execution in Heterogeneous Distributed Computing Environment

We consider the computational environment that includes two types of resources: dedicated (virtualized) and non-dedicated resources. Jobs from local users of clusters are operated directly by the RMS installed in non-dedicated resources.

We implemented job management using multi-agent technology. We developed MAS that includes various agents including agent-scheduler, agent-classifier, and agents that manage resources [23]. Each resource segment is represented by its agent. Fig. 1 shows a job execution scheme.

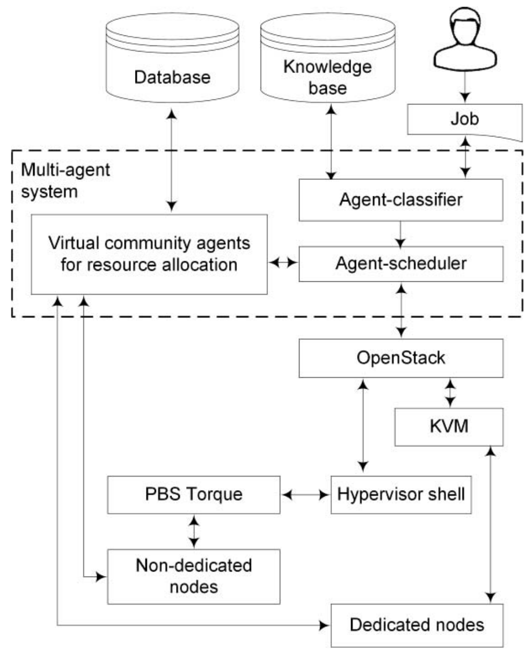

Figure 1. Job execution scheme.

The agent-classifier defines a job class based on a conceptual model of the environment [24]. When the job is classified, it is transferred to the agent-scheduler.

This agent finds resources that can execute the job of the defined class. These resources are represented by the Virtual Community (VC) of agents. The job is sent for execution into dedicated or non-dedicated resources. The formation of virtual communities enables agents to adapt the management process to the new challenges.

The problem of job management in distributed computing is NP-hard.

Our multi-agent algorithm obtains solutions applying economic mechanisms for regulation of resources supply and demand. We use a tender of computational work. It is based on the Vickrey auction model with one-round bidding [25].

We use the OpenStack platform for the resource virtualization and the KVM hypervisor for VMs provisioning. The OpenStack components are placed in the dedicated nodes, which are temporarily not part of the cluster.

The Portable Batch System (PBS) Torque system is used for clusters [26]. We consider its schedule as a set of slots. Each slot determines cores that are used or can be used during a certain time interval. The VM is launched in non-dedicated resources by a specially developed hypervisor shell.

If the dedicated resources are overloaded, and there are free slots in a schedule for RMS job queue, then the VM is launched using the RMS queue and executed on non-dedicated resources.

The agent-scheduler defines requirements for the job and resources taking into account the information about the current state of environment components. This information is elicited from a meta-monitoring system database [23]. If the job can be executed, then the agent-scheduler sends a request to the OpenStack platform for forming and launching the necessary number of VMs in non-dedicated nodes.

A VM launch scheme from the PBS Torque queue is shown in Fig. 2.

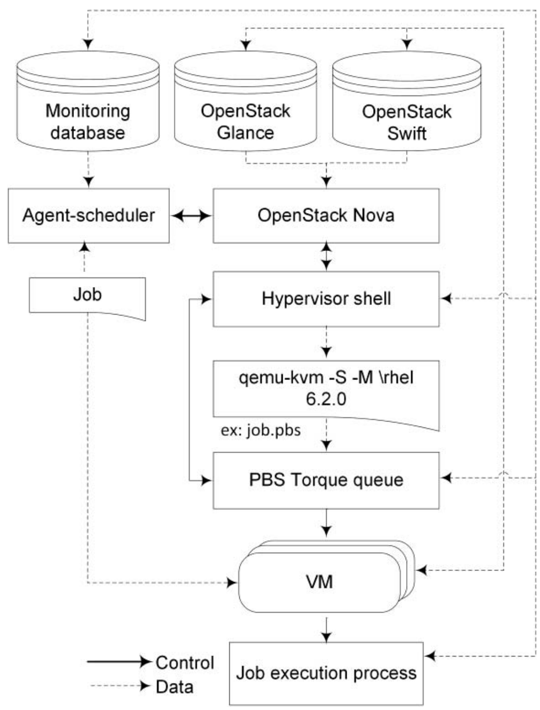

Figure 2. Scheme of job launch from the queue of RMS.

The following components of the OpenStack platform are used: scheduler Nova, cloud file storage Swift, and VM images library Glance. We use Nova to request VM and transfer it to the hypervisor shell using the standard protocol of the OpenStack platform.

The hypervisor shell operates in the control nodes of clusters together with the PBS Torque (Fig. 3). It creates a VM in the PBS Torque queue according to the received request.

The shell provides the following features: monitoring the current state of the PBS Torque queues to find released resources; adding system jobs for starting VMs to the selected queue; configuring launched VMs; monitoring the computations that are carried out in the VMs; VMs migration; VMs termination and resources release. The DIStributed COmputing system of Modular Programming (DISCOMP) framework carries out starting and managing jobs in VMs [27].

The PBS Torque receives the system job and pushes it into a queue. When the necessary number of resources is released, it executes the command for launching VMs on these resources. VMs images are stored in the Glance library. Network catalogs for VMs are mounted using information from the Swift storage that contains necessary data files and program libraries. When the VMs are configured and started, jobs are executed in them.

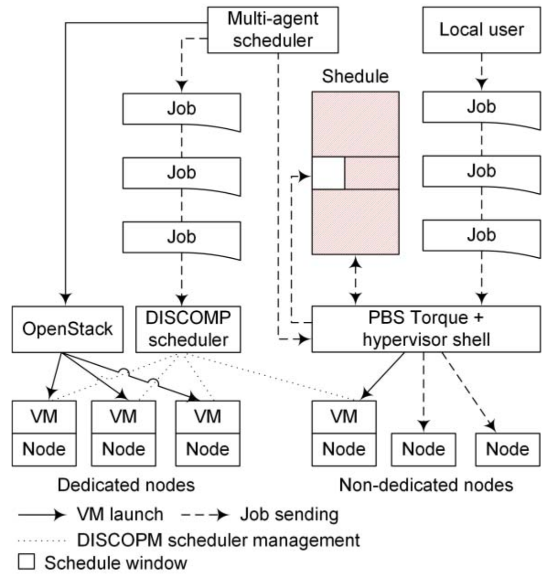

Figure 3. Scheme of adding a job for VM launch into a schedule slot.

IV. Experimental Analysis

In this section, we describe the aspects of our approach to the resource virtualization in the process of solving the complex practical problem. The objective is to study development directions of the Vietnam energy sector to optimize energy security for the defined period T.

We define the energy security as a protection of the country and national economy from the threats related to the insufficient provision of substantiated energy.

We study energy development taking into account all combinations of energy security threats. Such a large-scale study leads to a significant number of possible states and difficulties in the analysis by the classical methods [28].

In this regard, we apply combinatorial modeling methods. They are based on examining various combinations of energy sector states and transitions between them for the given period T.

The problem is represented by a directed graph $G = \langle V, U \rangle$, where V is a set of vertices (possible energy sector states at the studying period), and U is a set of edges (transitions between states). Edges are directed from the state s_0 at the initial time t_0 towards states $s_1, s_2, ..., s_n$ at the next time moments $t_1, t_2, ..., t_k$. Each vertex reflects the state of interconnected objects taking into account the unique set of parameter values.

Fig. 4 shows an example of the subgraph as a part of the energy sector. Fig. 5 describes the data associated with each graph's vertex. A typical power subsystem includes energy objects with different states: operated, reconstructed, modernized, and newly constructed objects. Each object performs one or more technological work (storing, production, processing, transformation, and transportation of energy resources).

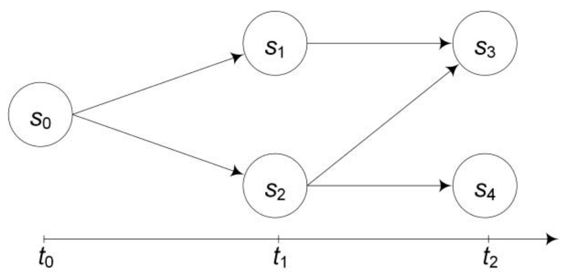

Figure 4. Example of the power subsystem development graph.

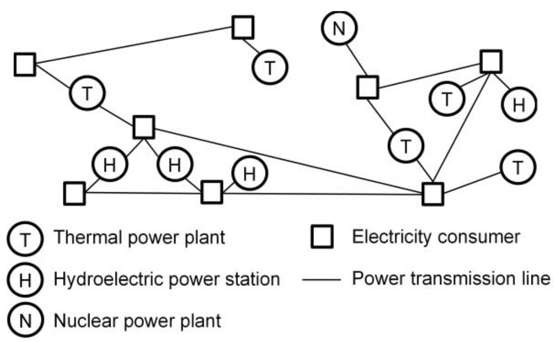

Figure 5. Scheme of the typical power subsystem.

The analysis is carried out by considering various combinations of subsystem states and transitions between them for the T [29].

The objective is to find the subgraph G_{opt} of the graph G with a minimum cost of development, functioning, and damage from energy resources shortage. We use the economic-mathematical model (1)-(4) of the energy sector:

$$F(x, y^1, y^2, ..., y^k) = (c, x) + \sum_{k=1}^{K} (g^k, r^k - y^k) \to \min , \quad (1)$$

$$Ax - \sum_{k=1}^{K} y^k = 0 , \quad (2)$$

$$0 \le x \le d , \quad (3)$$

$$0 \le y^k \le r^k . \quad (4)$$

The parameters have the following interpretation:

- c is the vector that represents costs per unit of technological work,
- x is the decision vector that shows the optimal amount of resources with various technologies per year,
- d is the vector that determines the upper limit of x,
- K is a number of consumer categories,
- k is the index of a consumer category,
- g^k is the vector that evaluates penalties per unit of the energy resource that is incompletely delivered,
- y^k is the decision vector that shows the optimal energy resource consumption,
- r^k is the vector that determines the upper limit of y^k,
- A is the matrix of coefficients that correspond technological chains from energy production resources to their transmission,
- Ax means the resource distribution over all chains.

Equation (2) determines the complete cycle of energy sector functioning.

We develop the distributed applied software package (scalable application) Corrective [30] for solving the problem. Corrective includes four modules.

The module m_1 defines the basic scenario of energy sector development (parameter z_4), based on the strategies of energy sector development (parameter z_1), critical situations (parameter z_2) and a set of energy security indicators (parameter z_3).

The module m_2 forms a parameter z_5, which aggregates a set of the possible energy sector states adjusted by sweeping its values in defined ranges.

The module m_3 is executed with different variants of its input parameter. The m_3 module carries out an admissibility evaluation of the energy sector state represented by the element $z_{5,i}$ of the parameter z_5. The obtained evaluation is represented by the element $z_{6,i}$ of the parameter z_6.

Finally, the module m_4 implements an expert analysis of energy sector states (elements $z_{6,1}, z_{6,2}, ..., z_{6,n}$ of the parameter z_6) from ensuring energy security point of view and forms the subgraph G_{opt} (parameter z_7) energy development.

All modules are developed as Microsoft Windows applications. The modules m_1, m_2, and m_4 are executed on remote nodes (personal computers).

The module m_3 is executed in virtualized resources functioning under control of GNU/Linux OS. We create a special image of VMs with the preset ReactOS operating system and m_3 [31]. This system provides compatibility with Microsoft Windows applications.

The problem solving is performed using the virtualized resources of the public access computer center "Irkutsk supercomputer center of Siberian Branch of the Russian Academy of Sciences". Five dedicated nodes of the high-performance cluster are used. Nodes have the following characteristics: 2x16 cores CPU AMD Opteron 6276, 2.3 GHz, 16 MB L3 cache, 4 FLOP/cycle, 64 GB RAM DDR3-1600. 160 VMs are started in these nodes. A transfer of data and control between the modules m_2, m_4, and m_3 are carried out by tools of the DISCOMP framework.

The non-dedicated resources include 55 nodes with characteristics that are similar to the dedicated node characteristics. They are managed by PBS Torque and used by users of the public access computer center.

We studied sustainable energy development of Vietnam from 2015 to 2030 on the basis of the model mentioned above. The main criterion is minimum of the cost of development and functioning of the energy sector. We determined the optimal subgraph G_{opt} of the graph G that consisted of 531442 states. The subgraph G_{opt} presents the scenario with increasing natural gas extraction and decreasing coal mining (Fig. 6). It is the better result than one received earlier with the heuristic evaluation of a rather small number of energy development variants.

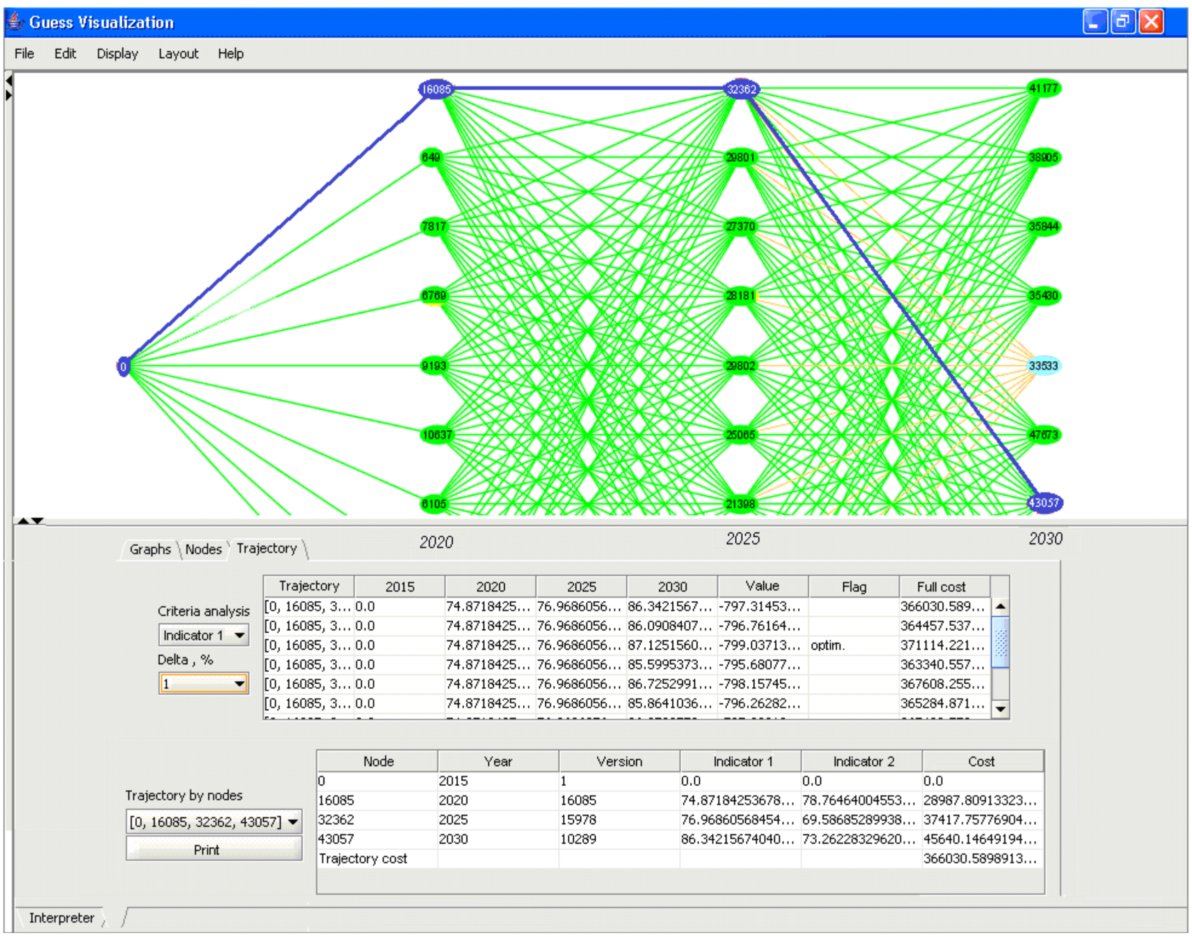

Figure 6. Optimal subgraph G_{opt}.

The module m_3 evaluates states every 10 seconds on average. Fig. 7 shows a distribution density of execution time of m_3 in seconds. These results show that the module m_3 execution time in most cases is about from 5 to 10 seconds. Therefore, the DISCOMP dispatcher provided a good balance of the of processor load for the used dedicated nodes.

The estimated time of problem-solving on one core is more than 60 days. The total time in the virtualized resources with 160 VMs is about 9 hours. A virtualization overhead of each of VMs did not exceed 1% of the total time of the work.

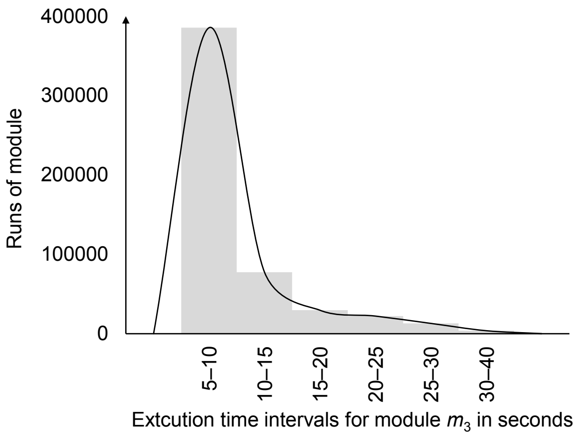

Figure 7. The distribution density of execution time for the module m_3.

A part of the computational load is re-allocated to non-dedicated nodes. Fig. 8 shows the number of free slots in the RMS schedule of non-dedicated resources and number of used ones for every hour in the solving process.

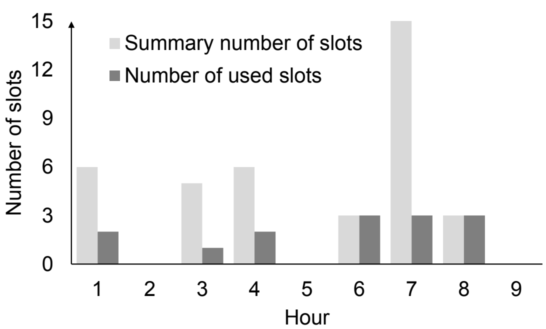

Figure 8. A number of slots vs. an hour.

Distribution densities of the number of cores and time intervals for the used slots are demonstrated in Fig. 9 and Fig. 10, respectively. We use the exponential fit to the histograms. Fig. 9 shows that most of the slots have 32 cores. At the same time, Fig. 10 demonstrates the insufficient time intervals in many slots for the effective launch and run of VMs.

We used 14 slots from 41 free slots. Six free slots occurred in the nodes in which jobs are already executed. The VMs startup could negatively affect the execution time. 21 slots exist short time. The efficiency of running machines in these slots would be extremely low.

In the best case, we use three slots at a time. The maximum number of additional cores is equal to 96. The standard deviation of the number of cores is equal 36.27. At the same time, the average processor utilization rate in non-dedicated resources is equal to about 90%.

Using the additional free slots by the agent-scheduler and hypervisor shell, we decrease the problem-solving time over 18% in comparison with OpenStack Nova-scheduler.

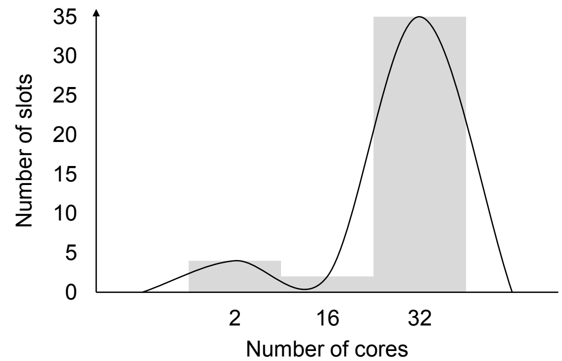

Figure 9. The distribution density of a number of cores in slots.

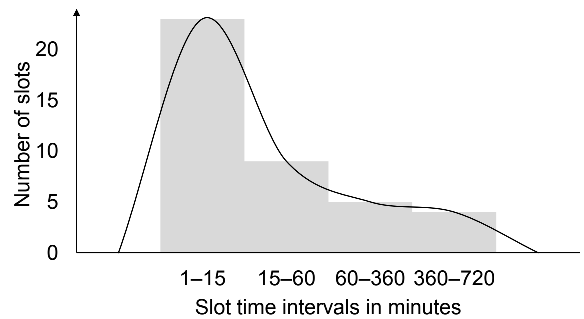

Figure 10. The distribution density of slot time intervals.

V. CONCLUSIONS

The dynamic elasticity of VM provisioning in a heterogeneous distributed computing environment has become a major issue for scalable systems. It allows handling the unpredictable workload and avoid overloading, but adds a new complexity dimension to the problem. The manner in which the resource provisioning can be done depends not only on the job property and resources it requires, but also users who share resources at the same time.

We consider a scenario, where computer infrastructure is divided into dedicated resources to support virtualized service and non-dedicated resources for traditional computations. We design the platform that supports dynamic management of non-dedicated resources when dedicated resources are overloaded. This approach provides dynamic elasticity of virtualized environment using the resources of both types. To this end, we developed the new multi-agent job scheduler for dedicated resources and special DISCOMP hypervisor shell in addition to the OpenStack platform.

We show advantages of this approach on a complicated practical problem related to the national energy development taking into account the energy security requirements. We solve it using parameter sweep application that demonstrates significant speed up of the problem-solving process. In addition, we demonstrate 18% of improvement by applying our elasticity mechanisms.

Acknowledgment

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

References

[1] Irkutsk Supercomputer Centre of SB RAS. [Online]. Available: http://hpc.icc.ru.

[2] V. K. Bumgardner, OpenStack in Action, Shelter Island: Manning Publications, 2016.

[3] S. Herres-Pawlis, A. Hoffmann, A. Balasko, P. Kacsuk, G. Birkenheuer, A. Brinkmann, L. de la Garza, J. Kruger, S. Gesing, R. Grunzke, G. Terstyansky, and N. Weingarten, "Quantum chemical meta-workflows in MoSGrid," Concurr. Comp. Pract. E., vol. 27, no. 2, pp. 344–357, February 2015.

[4] F. H. B. Megino, K. De, A. Klimentov, T. Maeno, P. Nilsson, D. Oleynik, S. Padolski, S. Panitkin, and T. Wenaus, "PanDA for ATLAS distributed computing in the next decade," J. Phys. Conf. Ser., vol. 898, no. 5, p. 52002, October 2017.

[5] B. Allen et al., "Globus: A Case Study in Software as a Service for Scientists," in Proceedings of the 8th Workshop on Scientific Cloud Computing, ACM, New York, 2017, pp. 25–32.

[6] S. S. Manvi and G. K. Shyam, "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey," J. Netw. Comput. Appl., vol. 41, pp. 424–440, May 2014.

[7] L. B. Sokolinsky and A. V. Shamakina, "Methods of resource management in problem-oriented computing environment," Program. Comput. Soft., vol. 42, no. 1, pp. 17–26, January 2016.

[8] G. Inggs, D. B. Thomas, and W. Luk, "A Domain Specific Approach to High Performance Heterogeneous Computing," IEEE T. Parall. Distr., vol. 28, no. 1, pp. 2–15, January 2017.

[9] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, and A. Y. Zomaya, "CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing," J. Grid Comput., vol. 14, pp. 23–39, March 2016.

[10] A. Tchernykh, U. Schwiegelsohn, E.G. Talbi, and M. Babenko, "Towards Understanding Uncertainty in Cloud Computing with risks of Confidentiality, Integrity, and Availability," J. Comput. Sci.-Neth, in press. [Online]. Available: http://dx.doi.org/10.1016/j.jocs.2016.11.011,

[11] R. Ramírez-Velarde, A. Tchernykh, C. Barba-Jimenez, A. Hirales-Carbajal, and J. Nolazco, "Adaptive Resource Allocation with Job Runtime Uncertainty," J. Grid Comput., vol. 15, no. (4), 415–434, December 2017.

[12] A. Singh and M. Malhotra, "Agent Based Framework for Scalability in Cloud Computing," Int. J. Comput. Sci. Eng. Technol., vol. 3, no. 4, pp. 41–45, April 2012.

[13] A. I. Kalyaev and I. A. Kalyaev, "Method of multiagent scheduling of resources in cloud computing environments," J. Comput. Sys. Sc. Int., vol. 55, no. 2, pp. 211–221, March 2016.

[14] V. V. Toporkov and D. M. Yemelyanov, "Economic model of scheduling and fair resource sharing in distributed computations," Program. Comput. Soft., vol. 40, no. 1, pp. 35–42, January 2014.

[15] B. Sharma, R. K. Thulasiram, P. Thulasiraman, and R. Buyya, "Clabacus: A Risk-Adjusted Cloud Resources Pricing Model Using Financial Option Theory," IEEE Trans. Cloud Comput., vol. 3, no. 3, pp. 332–344, July 2015.

[16] J. Hwang, S. Zeng, F. Wu, and T. Wood, "A component-based performance comparison of four hypervisors," in Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), IEEE, 2013, pp. 269–276.

[17] D. Ernst, D. Bermbach, and S. Tai, "Understanding the Container Ecosystem: A Taxonomy of Building Blocks for Container Lifecycle and Cluster Management," in Proceedings of the 2nd International Workshop on Container Technologies and Container Clouds, IEEE, 2016.

[18] S. Gupta and D. Gera, "A Comparison of LXD, Docker and Virtual Machine," Int. J. Sci. Eng. Res., vol. 7, no. 9, pp. 1414–1417, September, 2016.

[19] M. Plauth, L. Feinbube, and A. Polze, "A performance survey of lightweight virtualization techniques," Lecture Notes Comp. Sci., vol. 10465, pp. 34–48, September 2017.

[20] Apache CloudStack: Open Source Cloud Computing. [Online]. Available: https://cloudstack.apache.org.

[21] Euacalyptus. [Online]. Available: http://www.eucalyptus.com.

[22] OpenNebula. [Online]. Available: https://opennebula.org.

[23] I. V. Bychkov, G. A. Oparin, A. G. Feoktistov, I. A. Sidorov, V. G. Bogdanov, and S. A. Gorsky, "Multiagent control of computational systems on the basis of meta-monitoring and imitational simulation," Optoelectron. Instrum. Data Process., vol. 52, no. 2, pp. 107–112, March 2016.

[24] A. Feoktistov, A. Tchernykh, S. Gorsky, and R. Kostromin, "Knowledge elicitation in multi-agent system for distributed computing management," in 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, IEEE, 2017, pp. 1350–1355.

[25] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders," J. Financ., vol. 16, pp. 8–37, March 1961.

[26] Torque Resource manager. [Online]. Available: http://www.adaptivecomputing.com/products/open-source/torque/

[27] A. G. Feoktistov, I. A. Sidorov, V. V. Sergeev, R. O. Kostromin, and V. G. Bogdanova, "Virtualization of Heterogeneous HPC-clusters Based on OpenStack Platform," Bull. South Ural State Univ. Ser. "Computational Math. Softw. Eng., vol. 6, no. 2, pp. 37–48, May 2017.

[28] A. Herbst, F. A. Toro, F. Reitze, and E. Jochem, "Introduction to Energy Systems Modelling," Swiss J. Econ. Stat., vol. 148, no. II, pp. 111–135, December 2012.

[29] A. Edelev and I. Sidorov, "Combinatorial Modeling Approach to Find Rational Ways of Energy Development with Regard to Energy Security Requirements," Lecture Notes Comp. Sci., vol. 10187, pp. 317–324, April 2017.

[30] A. Edelev, V. Zorkaltsev, S. Gorsky, D. Van Binh, and N. H. Nam, "The Combinatorial Modelling Approach to Study Sustainable Energy Development of Vietnam," Commun. Comput. Inf. Sci., vol. 793, pp. 207–218, September 2017.

[31] ReactOS Project. [Online]. Available: https://reactos.org.