Thus, the article develops the ideology of solving the equations of mathematical physics, describing the transport phenomena. The diffusion sprawling of homogeneous concentration spots in an infinite horizontal tube has been considered. The probability density of the coordinates of the diffusing molecule has been put. The recursion relation for defining the statistical moments of the random variables with the help of diffusion equation has been obtained. It is shown that over time a normal distribution location is formed. Error at the replacement of the exact solution by its asymptotic expression is represented as the sum of polynomial amendments related to excesses given appropriate orders. The task of mathematical physics is reduced to a system of linear algebraic equations for the coefficients of the above mentioned polynomials. The structure of their solution is defined, according to which the coefficients are proportional to the number of their combinations, divided by the product of odd numbers, where the number of factors is determined by the degree of given coordinates. The coefficients of the proportionality of the following polynomial coefficients are connected with the previous ones by recurrence relation. Its consistent solution allows taking into account the diffusion contribution to the spread of the concentration spots of excesses of arbitrary order and with any accuracy required to construct a solution of the diffusion equation.

REFERENCES

1. Ехилевский, С.Г. Теоретико-вероятностный подход к моделированию динамической сорбционной активности / С.Г. Ехилевский, О.В.Голубева, Д.В. Пяткин // Вестн. Полоц. гос. ун-та. Сер. В, Промышленность. Прикладные науки. – 2013. – № 11. – С. 144–151.
2. Ехилевский, С.Г. Метод моментов и динамика сорбционной активности при малых временах / С.Г. Ехилевский, О.В. Голубева, С.А. Ольшаников // Вестн. Полоц. гос. ун-та. Сер. В, Промышленность. Прикладные науки. – 2013. – № 3. – С. 150–156.
3. Бугров, Я.С. Высшая математика / Я.С. Бугров, С.М. Никольский. – М. : Наука, 1989. – 464 с.
4. Гихман, И.И. Теория вероятностей и математическая статистика / И.И. Гихман, А.В.Скороход, М.И. Ядренко. – Киев : Вища школа, 1979. – 408 с.

UDC 005

VAGRANT AND ANSIBLE. DEVOPS METHODOLOGY IMPLEMENTATION

YAUHENI ZHYDZETSKI, YAUHEN SUKHAREU
Polotsk State University, Belarus

DevOps methodology requires the abolition of borders between developers and operating sector employers, which requires an implementation of specific techniques. One of them is an environment virtualization. In our company, we use Vagrant and Ansible tools for the virtualization.

The term "DevOps" is commonly used to describe a professional movement, which is in favour of a joint working relationship between developers and operating sector employers for a more rapid implementation of planned activities with increasing reliability, stability, sustainability and security of production-environment [1]. DevOps naturally inherits principles of flexible methodologies for the abolition of borders between developers, testers, administrators, quality specialists and other representatives of the operating sector, as well as for motivation to implement bold and risky cases with the fastest possible user's feedback.

Achievement of such goals requires the maximum automation of a working process: testing, continuous integration, continuous delivery, code quality check, creation of production-environments, monitoring. For realization of all these processes, there are many different solutions, each of them has its own pluses and minuses. According to the concept of DevOps, responsibility zones between various departments of technical sector of the company are washed away, that brings criterion of simplicity to the same level as functionality. The DevOps-professional must have an opportunity to write a code, write tests to it, create an environment similar to the one that will be at the production, change settings of this environment, constantly being under supervision of automatic quality control services and test covering. Of course, preventive annual preparation in a special center for a specialist isn't provided.

The problem of various environments at a developer, a tester and at a production-stand is known for everything. Its consequences are even better known: non-reproducible mistakes, the phrase "I don't know, everything works for me", unexpected falling of production-stands at excellent viability of service under the same loadings at test platforms, etc. Creation of identical configurations on 5-20 cars, it is in our century of cloudy services even not the average quantity, without automation of process turns into work that can be done only by a cool old school administrator.

Actually, the solution to all these problems is well-known and its name is a virtual machine. It is not a secret that networks of the majority of medium-sized and large companies are entirely constructed on virtual machines. This approach has a lot of advantages, the main of them is a low price and a good flexibility. With use of virtual machines it becomes simpler to distribute resources for various nodes of a network and to control access levels, to create new nodes, cloning virtual images of the existing machines.

The most widespread utile for management of virtualization of an environment of developer is Vagrant. Vagrant is a manager of virtual machines, it works with VirtualBox, VmWare and several other less popular systems of virtualization [3]. It has the simple console interface, and the main thing is working according to the scenario described in a configuration file with Ruby syntax. It means that one DevOps-specialist can write a scenario in this quite widespread programming language, place the file with the scenario under management of a version control system, and all team of developers working with it over the project will be able to use all the charm of virtualization by executing only one simple console command: *vagrant up*. If a developer can create the virtual computer with an environment for the developed project, he can alsocreate the same car both at the test stand and on the production-server, and it is already a step from the simple developer to the DevOps-specialist. Objectively, such approach gives the following advantages:

– repeatability of an environment. At a programmer, testers and at a production enviroments everything is identical;

– cleanness of a host-car. Work on various projects demands various environments and, if to put everything directly on the car, soon it will be a "zoo" from various DBMS, versions of interpreters, services and servers. When using a virtual machine, in host-car there is only a favorite IDE and some lightweight UI clients to the services working at the virtual machine;

– simplicity of connection of new people to the project. Big projects always have a set of external dependences, a new person will be able to start working and see a result of his actions, without studying a big set of documentation and configuration instructions in a project's wiki;

– it is easy to transfer from a tester to a developer a difficulty reproducing mistake for correction by a simple freeze of the virtual machine on the moment with a mistake and transfer the image.

The virtual machine during the work with Vagrant represents a certain initial image (a box) which is developed and configured. A big collection of basic images of systems is available in a Vagrant repository [4]. The configuration is presented in the broadest sense of this word: ports forwarding, control of general resources, software installation(provisioning). Software installation and configuration can be performed both at a low level: execution of shell-scripts – and with the use of instruments of management of configurations: Chef, Puppet, Ansible.

We chose Ansible as the simplest and rather functional instrument of management of configurations [2]. Ansible has almost zero threshold of entry for a programmer. All control of the target server, or group of servers, is performed through the SSH protocol. Opening additional ports and installation of additional software on the managed servers is not required.

Scenarios in Ansible are described by YAML language with the use of syntax of expressions of a Jinja2 template language. A scenario (playbook) specifies a configuration and a set of servers to which it is applied. Servers are presented in the so-called inventory-files: files in an ini-format in which are specified IP or DNS names of servers with a logical grouping: servers of databases, application servers, etc. As a rule on each environment: test, production – the inventory-file is created. The configuration consists of a list of commands, the ansible module with a list of parameters which is responsible for invoking each command. Technically, the module in Ansible is one instruction in a shell-. The main distinction is that the Ansible scenario has idempotency feature: if earlier a concrete change was applied, it won't be applied repeatedly – and it means that it is possible to bring improvements in the scenario and over and over again and apply them to the same set of machines. Ansible supports a set of modules and gives a possibility to write new own modules in Python language. A newer version of Ansible supports more modules then older version, for this reason it is always recommended to work at the last stable version.

Ansible allows to write all scripts in one playbook-file, in that case this file of the scenario represents the sequence of modules constructed by analogy with a simple shell-script. The scenarios written thus have a big size, they are difficult to support and aren't suitable for reuse at all. There is an easier way of execution of a command on the demanded set of servers – AdHoc [5]. This way doesn't require a file of a scenario at all, and can execute an instruction on the servers containing in the inventory-file from a command line. This style is extremely bad and it isn't recommended to use it at a production environment, because it allows to roll non-reproducible change on a set of servers. Any change which isn't recorded in the scenario under the version control system management is considered not reproduced as it won't be guaranteed applied to the servers added to the system in future so will lead to mistiming of system and create potential problems of support.

According to the best practices, it is recommended to group sets of changes into roles [6]. This term in Ansible is used to describe some established and adjusted software element: JRE, nginx-server, Tomcat, Mon-

goDB, etc. Creation of a role assumes a description of process of installation and configuration of concrete element, definition of a set of the possible parameters having impact on process, determination of dependences on other roles. The format of the description of a role is strictly regulated, the structure of a file system determined by the agreement.

For roles distribution there is a special repository with the name Ansible Galaxy [7]. Moderation of the roles published in this repository leaves much to be desired, there is a system of the user responses, but also it doesn't give quality assurance. Besides, Ansible does not have any cross-distributive features which creates additional difficulties in search of an existent role for application in its own environment. By trial and error method, we came to the tactics of our own realization of roles with the configurability, which is minimum demanded for reuse, and to their distribution through an internal corporate repository.

After usage of approach of grouping the configurations in roles, the file of the scenario turns into the simple list of roles and servers to which they have to be applied. Thus concrete servers for each environment are set in theirnown inventory-file. It is easy to expand and support a configuration of such format, it becomes similar to a good object-oriented code.

Nowadays, almost in all projects of our company the system of virtualization of an environment of the developer is used with the tools and approaches described above. As a result, wiki-instructions for new developers have practically disappeared, and the average time of the beginning of work on the new project has been considerably reduced. The degree of a working process compliance of methodology of DevOps includes a set of parameters and almost doesn't give in to an objective assessment, but a positive dynamics is obvious, so the correct reference point for progress is chosen.

REFERENCES

1 Шарма, С. Освоение концепции DevOps в интересах непрерывной поддержки инноваций / С. Шарма // IBM developerWorks [Электронный ресурс]. – 2014 – Режим доступа: https://www.ibm.com/developerworks/ru/library/d-devops-continuous-innovation. – Дата доступа: 25.09.2015.
2 Поляков, В. Ansible / В. Поляков // Habrahabr [Электронный ресурс]. – 2013 – Режим доступа: http://habrahabr.ru/post/195048. – Дата доступа: 26.09.2015.
3 Providers [Electronic resource] / Vagrant docs. – 2015 – Mode of access: https://docs.vagrantup.com/v2/providers. – Date of access: 25.09.2015.
4 Vagrant boxes index [Electronic resource] / Vagrantbox.es. – 2015 – Mode of access: http://www.vagrantbox.es. – Date of access: 25.09.2015.
5 Introduction To Ad-Hoc Commands [Electronic resource] / Ansible documentation. – 2015 – Mode of access: http://docs.ansible.com/ansible/intro_adhoc.html. – Date of access: 26.09.2015.
6 Best Practices [Electronic resource] / Ansible documentation. – 2015 – Mode of access: http://docs.ansible.com/ansible/playbooks_best_practices.html. – Date of access: 26.09.2015.
7 Welcome to Ansible Galaxy [Electronic resource] / Ansible Galaxy. – 2015 – Mode of access: https://galaxy.ansible.com/intro. – Date of access: 26.09.2015.

UDC 004.9.23

THE GRAPH VISUALIZATION PROBLEMS OF ARTIFICIAL INTELLIGENCE

TATYANA RUDKOVA, MARIA DEKANOVA
Polotsk State University, Belarus

The article deals with the criteria for quality visualization of graphs for various subject areas, as well as algorithms for graph visualization in accordance with predetermined criteria. Scroll to the following criteria: the number of intersections of the ribs, the unification of edge lengths, the area of occupancy, number of folds, symmetry.

Currently, research in these areas traditionally belongs to mathematics taking a more prominent place. The problem of choosing the optimal solution for various kinds of applications, is one of the most pressing technical and economic problems [1].

Development of the theory of graphs in the main obligation to the large number of various applications, such as scheduling the movement of vehicles, placing items ambulance or telephone stations, optimal selection of the intensities of work, the problem of the distribution of work, placement of dispatch centres of urban transport networks as well as network design, television broadcasting and other [2,3]. Therefore, the graphs have been used in virtually every branch of science: physics, biology, chemistry, mathematics, history, linguistics, social sciences, technology, etc. The most popular graph-theoretic model is used in the study of communication net-