

Суперкомпьютерный консорциум университетов России  
Российская академия наук



# Суперкомпьютерные дни в России

Труды международной конференции

21–22 сентября 2020 г., Москва



---

МОСКВА – 2020

УДК 519.7  
ББК 22.18  
С89

*Конференция проведена при поддержке  
Российского фонда фундаментальных исследований (проект 20-07-22003)*

Под редакцией члена-корреспондента РАН *Вл. В. Воеводина*

**С89** **Суперкомпьютерные дни в России** : Труды международной конференции. 21–22 сентября 2020 г., Москва / Под. ред. Вл. В. Воеводина. – Москва : МАКС Пресс, 2020. – 172 с.

ISBN 978-5-317-06400-6                      e-ISBN 978-5-317-06458-7  
<https://doi.org/10.29003/m1406.RussianSCDays-2020>

Данный сборник содержит полные статьи на русском языке, короткие статьи, аннотации стендовых докладов и аннотации полных англоязычных статей, включенных в программу Международной конференции «Суперкомпьютерные дни в России».

УДК 519.7  
ББК 22.18

**Russian Supercomputing Days: Proceedings of the International Conference. September 21–22, 2020** / Ed. by Vl. Voevodin. – Moscow : MAKS Press, 2020. – 172 p.

ISBN 978-5-317-06400-6                      e-ISBN 978-5-317-06458-7  
<https://doi.org/10.29003/m1406.RussianSCDays-2020>

The book contains full papers in Russian, short papers, poster abstracts, and also abstracts of full papers in English included in the “Russian Supercomputing Days” conference agenda.

*Подробную информацию о конференции можно найти в сети Интернет  
по адресу <https://RussianSCDays.org>*

ISBN 978-5-317-06400-6  
e-ISBN 978-5-317-06458-7  
<https://doi.org/10.29003/m1406.RussianSCDays-2020>

© Авторы статей, 2020  
© МГУ имени М.В. Ломоносова, 2020  
© Оформление. ООО «МАКС Пресс», 2020



**Полные и короткие статьи**

# A Software Package for Studying Dynamical Systems Using the Lyapunov Characteristic Exponent in a Hybrid Computing Environment\*

S.V. Vostokin , A.V. Doroshin

Samara National Research University

The article proposes a method for the numerical analysis of dynamical systems and a software package for its implementation. The mathematical basis of the method is the calculation of the Lyapunov characteristic exponents for trajectories of dynamical systems in phase space. Parallelization of the computational process is performed by replenishing the list of running tasks for calculating Lyapunov characteristic exponents depending on the result of completed tasks at the pace of calculations. To automate a computational experiment, we developed and tested a distributed application. The integration of application components is based on the principles of an enterprise desktop grid and cloud systems. A feature of the application architecture is the deployment in a hybrid computing environment based on academic and free cloud systems using the MyBinder and Everest cloud platforms together with JupyterLab and Maple packages.

*Keywords:* Lyapunov characteristic exponent · Dynamical system · Hybrid cloud · Enterprise desktop grid

## 1 Introduction

At present, the study of the processes of nonlinear dynamics and the chaotic behavior of complex systems based on the numerical models is gaining more and more interest. From a practical point of view, these studies are the basis for the development of complex control systems. For example, the numerical integration of differential equations is used in the developing spacecraft orientation and control systems. From a theoretical point of view, it is interesting to study the mathematical properties of differential equations. For example, for a given system of differential equations, numerical analysis allows you to build maps of phase states and phase transitions. In addition, numerical or combined numerical-analytical methods are often a non-alternative research tool in the subject area under consideration. The reason is the difficulty or inability to obtain analytical solutions for the complex equations.

At the same time, in the field of dynamical systems modeling, there are a number of technical problems that make it difficult to use numerical methods of research.

Firstly, the complexity of a computational experiment is high. Even finding one solution with acceptable accuracy on modern computers requires significant computational costs. But for a detailed analysis, it is necessary to find a large number of solutions with different initial conditions and settings of the numerical method.

Secondly, a combination of both numerical and analytical instrumental approaches is required. Such a combination is necessary since manual programming of the numerical method is time-consuming. This, on the one hand, leads to unproductive expenditures of the researcher's time. On the other hand, it can lead to software implementation errors and incorrect results.

Thirdly, methods for studying a phase spaces based on strict search algorithms and strict criteria for detecting the desired behavior of a dynamical system are rarely used. Instead, the search is based on the intuition and experience of the researcher. The subjective analysis of visual modeling artifacts is used. The example is the manual phase diagram analysis. This affects the reproducibility of the results and the completeness of the study.

The first problem does not allow studying dynamical systems on one computer: high-performance computing technologies are required. The second problem creates difficulties when applying tradition-

---

\* The work is supported by the Russian Science Foundation (# 19-19-00085)

al technologies of high-performance computing on clusters, namely MPI and OpenMP. The third problem requires the development of a convenient technology for managing computational experiments.

The paper proposes a solution to these problems, which is based on a combination of proven technologies for building enterprise-level desktop grid systems [1] and modern web-based computer modeling systems. The last mentioned systems developed rapidly following the popularity of machine learning, data processing, and artificial intelligence. We also study the use of hybrid academic and free cloud systems [2] for the deployment of our software package.

The paper has the following structure. In the Section 2, we propose an implementation of the Lyapunov characteristic exponent method, which underlies the analysis procedures of dynamical systems. In the Section 3, we consider the architecture of the software package, the interaction of its components and the method of deployment to computing resources. In the Section 4, using load and integration tests, we examine the performance of the software package. In the Conclusions, an overview of the work is given and directions for further research are indicated.

## 2 Implementation of the Lyapunov Characteristic Exponent Method

In the section we briefly consider the method for calculation of Lyapunov characteristic exponents for Lorentz system [3]. The calculation can be done using the well-known Benettin's algorithm with Gram-Schmidt orthogonalization [4].

The problem of calculating Lyapunov characteristic exponents is based on the numerical integration of the differential equations. In the case of Lorenz system we have the following equations:

$$\dot{x} = \sigma(y - x); \dot{y} = rx - y - xz; \dot{z} = -bz + xy. \quad (1)$$

To obtain the Lyapunov characteristic exponents we need to integrate the system (1) and the auxiliary systems for variations, namely the three subsystems for variations  $\{\dot{x}_i, \dot{y}_i, \dot{z}_i\}, i = 1, 2, 3$ :

$$\begin{cases} \dot{x}_1 = \sigma(y_1 - x_1); \dot{y}_1 = rx_1 - y_1 - x_1z - xz_1; \dot{z}_1 = -bz_1 + x_1y + xy_1; \\ \dot{x}_2 = \sigma(y_2 - x_2); \dot{y}_2 = rx_2 - y_2 - x_2z - xz_2; \dot{z}_2 = -bz_2 + x_2y + xy_2; \\ \dot{x}_3 = \sigma(y_3 - x_3); \dot{y}_3 = rx_3 - y_3 - x_3z - xz_3; \dot{z}_3 = -bz_3 + x_3y + xy_3. \end{cases} \quad (2)$$

The integration procedure is implemented for all twelve equations (1-2). At each integration step (at time step  $T$ ), normalization and orthogonalization of the variation vectors are performed. The partial sum

$$S_i = \sum_{k=1}^N \ln \|\{x_i(T), y_i(T), z_i(T)\}\| \quad (3)$$

is calculated during the integration procedure. At the end of the procedure, the sum  $S_i$  will be equal to Lyapunov characteristic exponents.

The code for the integration procedure is presented below in the format of a Maple worksheet. The code implements Benettin's algorithm with Gram-Schmidt orthogonalization for Lorentz system with given values of  $b, r$ , and  $\sigma$ .

```
with(plots):
with(DEtools):
with(plottools, line):with(linalg):
b:=8/3: r:=28: sigma_:=10:
```

The Lorentz system is presented in the system of equation  $ur1, ur2, ur3$ .

```
ur1:=diff(x(t), t)=-sigma_*(x(t)-y(t)) :
ur2:=diff(y(t), t)=(r*x(t)-y(t)-x(t)*z(t)) :
ur3:=diff(z(t), t)=(x(t)*y(t) -b*z(t)) :
```

Three systems in variations:

```

ur11:=diff(x1(t),t)=-sigma_*(x1(t)-y1(t)) :
ur21:=diff(y1(t),t)=(r*x1(t)-y1(t)-x(t)*z1(t) -x1(t)*z(t) :
ur31:=diff(z1(t),t)=(x1(t)*y(t) +x(t)*y1(t) -b*z1(t)) :
ur12:=diff(x2(t),t)=-sigma_*(x2(t)-y2(t)) :
ur22:=diff(y2(t),t)=(r*x2(t)-y2(t)-x(t)*z2(t) -x2(t)*z(t) :
ur32:=diff(z2(t),t)=(x2(t)*y(t) +x(t)*y2(t) -b*z2(t)) :
ur13:=diff(x3(t),t)=-sigma_*(x3(t)-y3(t)) :
ur23:=diff(y3(t),t)=(r*x3(t)-y3(t)-x(t)*z3(t) -x3(t)*z(t) :
ur33:=diff(z3(t),t)=(x3(t)*y(t) +x(t)*y3(t) -b*z3(t)) : .

```

Initial conditions for phase trajectory representing co-called strange attractor.

```

x0:=0.5:y0:=1:z0:=1.5:
T:=0.05:Num_of_steps:=10000:LAP1:=vector(Num_of_steps,[]):LAP2:=vector(Num_of_steps,[]):LAP3:=vector(Num_of_steps,[]):
epsilon_res:=1:

```

Initial conditions for systems in variations:

```

nach_01:=vector(3,[epsilon_res,0,0]):
nach_02:=vector(3,[0,epsilon_res,0]):
nach_03:=vector(3,[0,0,epsilon_res]):
S1:=0:S2:=0:S3:=0: .

```

Here is the beginning of the integration procedure.

```

for i from 1 by 1 to Num_of_steps do:
Resh1:=dsolve(
[ur1,ur2,ur3,ur11,ur21,ur31,ur12,ur22,ur32,ur13,ur23,ur33,x(0)=x0,y(0)=y0,z(0)=z0,x1(0)=nach_01[1],y1(0)=nach_01[2],z1(0)=nach_01[3],x2(0)=nach_02[1],y2(0)=nach_02[2],z2(0)=nach_02[3],x3(0)=nach_03[1],y3(0)=nach_03[2],z3(0)=nach_03[3]],
[x(t),y(t),z(t),x1(t),y1(t),z1(t),x2(t),y2(t),z2(t),x3(t),y3(t),z3(t)],
type=numeric, method=rkf45,relerr=0.000001,abserr=0.000001, output=listprocedure):
X1 := eval(x1(t),Resh1):Y1 := eval(y1(t),Resh1):Z1 := eval(z1(t),Resh1):
X2 := eval(x2(t),Resh1):Y2 := eval(y2(t),Resh1):Z2 := eval(z2(t),Resh1):
X3 := eval(x3(t),Resh1):Y3 := eval(y3(t),Resh1):Z3 := eval(z3(t),Resh1):

```

The numerical solution based on the integration procedure is called *Resh1*:

```

X := eval(x(t),Resh1):Y := eval(y(t),Resh1):Z := eval(z(t),Resh1):
xyzT:=vector(3,[X(T),Y(T),Z(T)]):
xyz1T:=vector(3,[X1(T),Y1(T),Z1(T)]):
xyz2T:=vector(3,[X2(T),Y2(T),Z2(T)]):
xyz3T:=vector(3,[X3(T),Y3(T),Z3(T)]): .

```

Doing normalization, orthogonalization and calculating of the partial sums *S<sub>i</sub>*:

```

norm_xyz1T:=norm(xyz1T):
S1:=S1+ln(norm_xyz1T/epsilon_res):
xyz1_0:=evalm(xyz1T/norm_xyz1T):
xyz2_1:=xyz2T-evalm(multiply(xyz2T,xyz1_0)*xyz1_0):
xyz2_0:=evalm(xyz2_1/norm(xyz2_1)):

```

```
S2:=S2+ln(norm(xyz2_1)/epsilon_resh):
xyz3_1:=xyz3T-evalm(multiply(xyz3T,xyz1_0)*xyz1_0)
      evalm(multiply(xyz3T,xyz2_0)*xyz2_0):
xyz3_0:=evalm(xyz3_1/norm(xyz3_1)):
S3:=S3+ln(norm(xyz3_1)/epsilon_resh): .
```

New initial conditions for systems on the next iteration:

```
nach_01:=xyz1_0:nach_02:=xyz2_0:nach_03:=xyz3_0:
x0:=X(T):y0:=Y(T):z0:=Z(T): .
```

Fixing the current values of Lyapunov characteristic exponents and finishing the iteration.

```
LAP1[i]:=S1/(i*T):LAP2[i]:=S2/(i*T):LAP3[i]:=S3/(i*T):
end do:
```

The triple of real numbers  $\{LAP1[Num\_of\_steps]; LAP2[Num\_of\_steps]; LAP3[Num\_of\_steps]\}$  is the main output of the task in our software package for studying dynamical systems. The test runs of the Maple worksheet give the correct values of Lyapunov characteristic exponents  $\{+0.88; 0; 14.4\}$ . The dependencies of the values on the number of time-step are presented in Fig.1. The functions  $S_i(t)$  in Fig. 1 are also correspond show the correctness of the calculation.

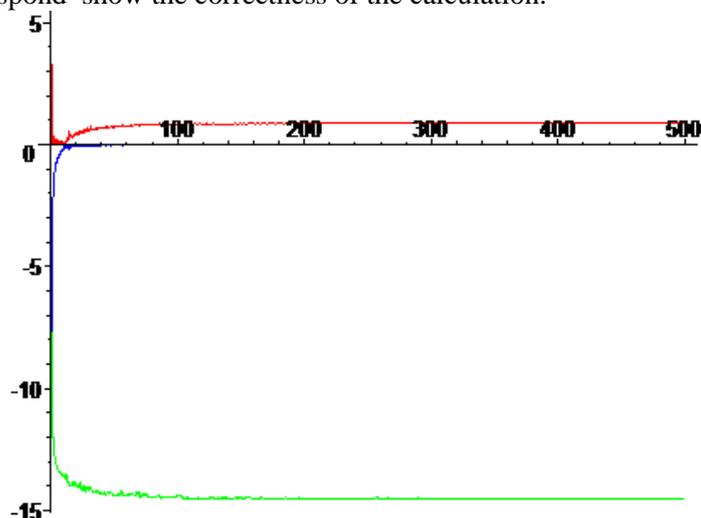
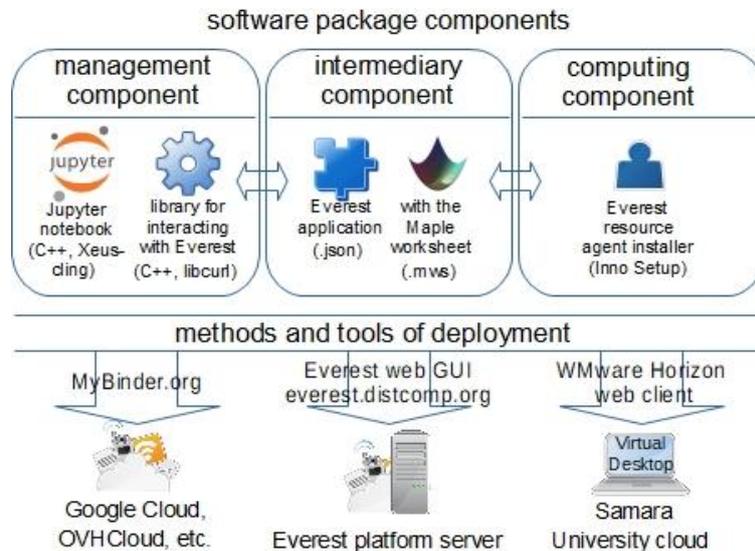


Fig. 1. The time-history of Lyapunov characteristic exponents.

### 3 Architecture of the Software Package

Let's consider the proposed architecture of the software package based on its deployment components (Fig. 2). We use three deployment components: (a) a management component, (b) an intermediary component; (c) a computing component.



**Fig. 2.** Architecture of the software package.

The management component is used to prepare tasks that calculate Lyapunov characteristic exponents for various initial conditions of the dynamical systems under study. In turn, it consists of two parts: a Jupyter notebook with code for generating tasks which calculate Lyapunov characteristic exponents; a library for interacting with the Everest platform [5] server using the REST protocol. Jupyter notebook code and the library are implemented in C++. The libcurl library is used for networking. Jupyter notebook uses the Xeus-cling kernel to execute C++ code. A feature of our parallel task-based [6] implementation is the ability to replenish the list of tasks launched through the Everest server at the pace of calculations.

The management component is deployed in conjunction with the JupyterLab [7] server in the MyBinder [8] cloud service. Deployment is performed on a free virtual machine from a repository stored on the GitHub service. The MyBinder service can randomly choose public clouds such as Google Cloud, OVHCloud, etc. for the deployment. After completion of the automatic deployment procedure, the researcher works interactively with the management component of the software package via a web browser.

The intermediary component is a special application that is pre-installed on the Everest server. The installation process is performed once for a series of experiments through a web graphical user interface (web GUI) provided by the Everest platform.

The Everest application defines how to process the task parameters passed from the management component to the Everest server in order to call the task on the computing component. The Everest application also defines the method of transmitting the result of the task from the computing component to the management component. The Maple worksheet that computes Lyapunov characteristic exponents is uploaded to the Everest server as a part of the application.

The computing component is represented by the Everest resource agent. We developed the installer program (based on Inno Setup) to setup the agent on a computing resource. After the installation, the agent is registered on the Everest server. Both ordinary workstations and virtual machines with pre-installed Maple package can be used as computing resources. We have the VMware virtual desktop service at our disposal (VMware Horizon web client). The service allows us to run virtual machines from a web browser.

Thus, the software package is launched and configured completely through a web browser. The researcher works with the software package in the familiar paradigm of the REPL (read-eval-print loop) interface, supported by both Maple and JupyterLab.

## 4 Discussion of Integration and Load Testing

In computational experiments, we (a) measured the typical execution time of one task of computing Lyapunov characteristic exponents; (b) evaluated the impact of task management overhead on the efficiency of computing; (c) examine the possibility of performing a series of experiments without failures and the correctness of the interaction between the components. The main experimental results of load testing are given in Table 1.

**Table 1.** Experimental results of load testing.

task delay, s	$T_1$ , s	$T_p$ , s	$S = T_1/T_p$	$E = T_1/(T_p \cdot p)$
total number of tasks = 100, number of task slots $p = 100$ , number of VM = 1				
20	2000	39.5779	50.5333	0.505333
80	8000	106.36	75.2163	0.752163
320	32000	345.096	92.7278	0.927278
total number of tasks = 100, number of task slots $p = 25$ , number of VM = 1				
20	2000	103.534	19.3174	0.772696
80	8000	347.421	23.0268	0.921073
320	32000	1302.87	24.5611	0.982445
total number of tasks = 400, number of task slots $p = 100$ , number of VM = 4				
20	8000	125.667	63.6606	0.636606
80	32000	387.256	82.6327	0.826327
320	128000	1352.49	94.6404	0.946404

The typical calculation of Lyapunov characteristic exponents on Intel Core i7-4510u 2.6 GHz computer (4 Gb, 64-Windows10) with Maple 10 installed took approximately 270 seconds (4.5 minutes). It means that carrying out a series of experiments really requires parallelization, and the time order for solving single task is typical for enterprise desktop grid systems.

We applied the following technique for software emulation of the operation of our grid system with several computing components. The task of computing Lyapunov characteristic exponents on the computing resource was emulated by a code with controlled task delay. Parameters were transferred to dummy tasks and a results was received. But instead of computing, the task process was put into standby by calling *Sleep(time)* function for a given *time*. The settings of the Everest agent on the computing resource were made in such a way that the Everest server distributed several tasks (from 25 to 100) to one resource at once. The actual runtime  $T_p$  of the application was measured on a series of dummy tasks for  $p$  simultaneously running tasks. Taking into account the total delay time  $T_1$  in all the tasks, the speedup was estimated as  $S = T_1/T_p$ , and the efficiency as  $E = T_1/(T_p \cdot p)$ . Thus, the load on the management component of our software package and the network overheads were estimated.

The efficiency of computing Lyapunov characteristic exponents in our method is at least 50% in the worst case (for task delay = 20 seconds, number of virtual machines VM = 1, number of task slots  $p = 100$ ) and near to 90% for the estimated typical time (270 seconds) in all studied configurations. This indicates good scalability of the proposed architecture, corresponding to [9]. The scalability depends on the number of task slots, since the control component must poll the status of all task slots in separate requests. However, the experiments shows that the software package can be used to run on at least 100 virtual machines with one task slot on each virtual machine.

We also made sure that despite the hybrid nature of the computing system with geographically distributed components, it is guaranteed to complete the calculation of 400 tasks (number of VM = 4, number of task slots  $p = 100$ ) within 20 minutes without failures. Experiments with the same deployment design show even higher reliability [10]. The noted properties indicate the applicability of the software package.

## 5 Conclusions

The article proposes a method for studying the behavior of dynamical systems based on the calculation of Lyapunov characteristic exponents. A software package was developed that automates resource-intensive computational experiments using this method. The novelty of the method, which pro-

vides advantages over traditional approaches, is the adaptation of enterprise desktop grid computing and modern approaches to computing, including web systems and cloud web services. In integration and load tests, we showed the applicability of the proposed method.

In the future, we plan to use this method to develop specific algorithms for the analysis of dynamical systems.

## References

1. Ivashko, E., Chernov, I., Nikitina, N.: A survey of desktop grid scheduling. *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, No. 12, pp. 2882-2895. IEEE (2018).
2. Goyal, S.: Public vs private vs hybrid vs community-cloud computing: a critical review. *International Journal of Computer Network and Information Security*, vol. 6, No. 3, p. 20. MECS Press (2014).
3. Lorenz, E. N.: Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141. AMS (1963)
4. Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.-M.: Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems: A method for computing all of them. P.I: Theory. P.II: Numerical application. *Meccanica*, vol. 15, pp.9-30. Springer (1980).
5. Sukhoroslov, O., Volkov, S., Afanasiev, A. A.: Web-Based Platform for Publication and Distributed Execution of Computing Applications. 14th International Symposium on Parallel and Distributed Computing (ISPDC), pp. 175-184. IEEE (2015).
6. Thoman, P., et al.: A taxonomy of task-based parallel programming technologies for high-performance computing. *The Journal of Supercomputing*, vol. 74, No. 4, pp. 1422-1434. Springer (2018).
7. Granger, B., Colbert, C., Rose, I.: JupyterLab: The next generation Jupyter frontend. *JupyterCon 2017*, New York (2017).
8. Jupyter, P., Bussonnier, M., Forde, J., Freeman, J., Granger, B., Head, T., Holdgraf, C., Kelley, K., Nalvarte, G., Osheroff, A., Pacer, M., Panda, Y., Perez, F., Ragan-Kelley, B., Willing, C.: Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. *Proceedings of the 17th Python in science conference*, vol. 113, p. 120. Austin, Texas (2018).
9. Senger, H., da Silva, F. A. B.: Bounds on the scalability of bag-of-tasks applications running on master-slave platforms. *Parallel Processing Letters*, vol. 22., No. 02, p. 1250004. World Scientific (2012).
10. Vostokin, S.V., Sukhoroslov, O.V., Bobyleva, I.V., Popov, S.N.: Implementing computations with dynamic task dependencies in the desktop grid environment using Everest and Templet Web. *CEUR Workshop Proceedings, Volume 2267*, pp. 271-275. CEUR-WS, Dubna (2018).

# SYSTEM OF INTELLIGENT SUPPORT FOR SOLVING GEOPHYSICAL PROBLEMS<sup>1\*</sup>

B.M. Glinskiy, G.F. Zhernyak, P.A. Titov

Institute of Computational Mathematics and Mathematical Geophysics SB RAS

*Annotation:* In this paper, we consider the system of intelligent support (SIS) for solving geophysical problems, that is being developed by the authors at the Institute of Computer Science and Technology of the Siberian Branch of the Russian Academy of Sciences. The usefulness of such SIS is substantiated. The proposed system is based on the ontology of parallel supercomputer architectures and the ontology of methods and algorithms available. Ontology is understood as a system of concepts and their relations. For clarity, illustrations are presented that schematically display the various components of the process of solving a geophysical problem, from stating to implementation on a supercomputer, and also how those are interconnected. Two different problems are described via means of SIS: 3D seismic wave field modeling and reconstruction of seismic section via pre-stack time migration. An example of field observation processing for one of the areas of Western Siberia using the developed system is presented.

*Key words:* Supercomputers, Geophysics, Intelligent support of computations, Ontology, Seismic, Velocity profile, Modeling

## 1. Introduction

Each researcher in any subject area related to the application of supercomputer technologies faces the questions: 1) what method (set of methods) to use for solving the problem? 2) what parallel architecture to use for carrying out numerical calculations? He needs to solve a specific problem on a supercomputer in an acceptable time, with sufficient accuracy.

The researcher should be well oriented in all these areas, which is not always manageable. Since, as a rule, if one is well versed in methods for solving the problem, he is less versed in modern supercomputer architectures and supercomputer technologies, and vice versa. The creation of SIS for decision making, linking all the necessary components, is aimed to solve such a problem. The user of the system will get a visual representation of the available methods for setting and solving the problem and how it relates to supercomputer technologies implementation.

The presentation of knowledge using the ontological approach has been growing in popularity over the past few years [1,2]. The need to create such systems was put forward in different fields of science as geology [3], physics [4], astrophysics [5,6,7], biology [8].

Thus, on the one hand, there is a set of methods, algorithms, and, possibly, program codes for solving the task, and on the other hand, there is a set of parallel architectures, various program libraries and tools. Gathering in one place the existing knowledge about a specific subject area (what are the tasks, solution methods, tools, and also describe the interactions) is an urgent task. The system created is a “living” object. One can constantly add new components and new connections to it.

At the moment, it is not known about the existence of open resources with an ontological representation of methods for solving geophysical problems on supercomputers, as well as open resources with an ontological representation of geophysical observation data. As part of the work, the authors present their vision for SIS development for solving geophysical problems.

---

<sup>1\*</sup> This research was supported by budget project 0315-2019-0009 (sections 1), by the Russian Foundation for Basic Research [grants No. 19-07-00085] (sections 2, 3, 3.2), by the Russian Foundation for Basic Research [grants No. 20-01-00231] (section 3.1).

## 2. Basic principles of SIS

The concept of the proposed intellectual support system for solving compute-intensive problems is described in [6,7]. Figure 1 shows a diagram that makes it easier to understand about SIS objects at the upper level and how different components are linked (the figure is taken from [7]).

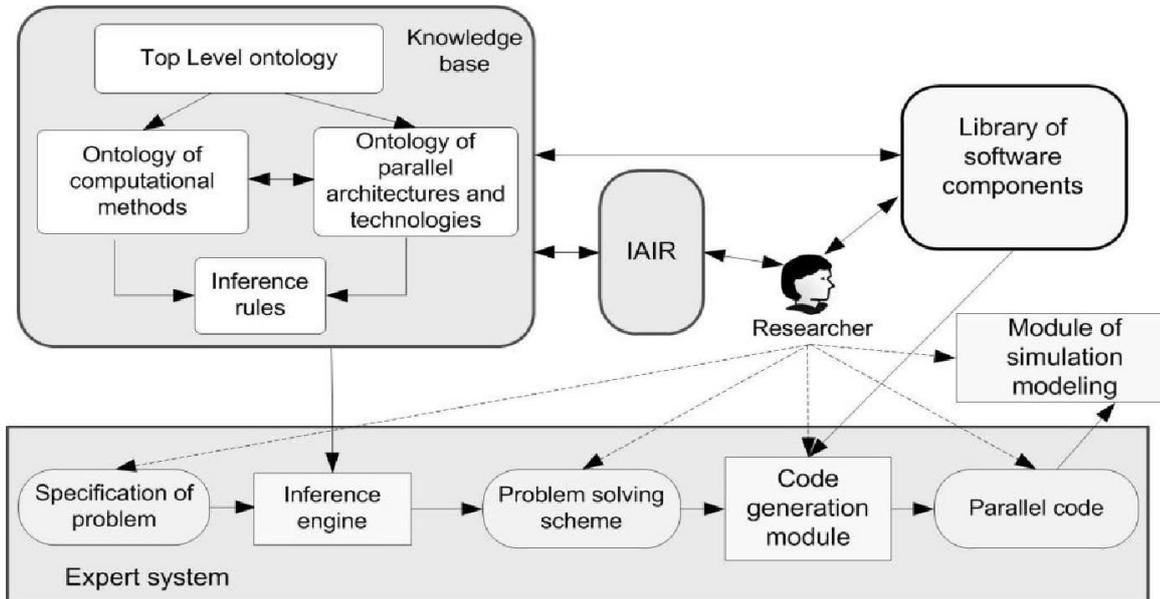


Fig. 1. Diagram of SIS for solving compute-intensive problems

The system is framed around knowledge base in which the available ontologies of computational methods, parallel architectures and technologies are placed. The logic of these ontologies is expanded with inference rules.

The inference engine (resolver) uses the knowledge base and the specification module of compute-intensive problem (compiled by the user) to construct the most optimal way for solving the problem. The code generation module is responsible for supporting the creation of optimal parallel code by offering the necessary libraries and parallel programming technologies.

The software components library includes code fragments that implement the necessary algorithms that are executed on a supercomputer. Software components are provided with unified specifications, on the basis of which they can be integrated into user's code. Also, the PC library provides the user with complete software packages that he can apply in the course of solving his task.

The information and analytical Internet resource module supports the solution of computationally complex problems on supercomputers (IAIR). It provides complete information about the methods and algorithms available, about the possibilities and limitations of each of them, about the characteristics of their implementations. This resource is equipped with a developed web-based user interface that provides meaningful access to this kind of information. The user will be able to obtain detailed information necessary to solve his problem: what methods and algorithms are currently available, their description and features of numerical implementation; available tools for creating parallel code; available supercomputer architectures and their features, as well as programming features on these architectures. Because of such capabilities of the IAIR, the user can significantly reduce the time required for in-depth familiarization with the problem area, since all the necessary information is structured and collected in one place.

A simulation block is used to evaluate the scalability of parallel code on multi-core systems. Using it, one can get temporal estimate of code execution on a different number of cores for a given parallel architecture, thereby choosing the best solution for the user. This approach was first used for the problem of the propagation of seismic waves in volcanic structures [9] and for modeling 3D seismic fields in areas with complex surface topography [10].

For a more detailed description of the computational methods ontology and the ontology of parallel architectures and technologies, one can refer to [6]. Here we only note that in practice ontologies are described in the OWL language [11]. This includes a description of classes with its properties, as well as elements of these classes that fill the knowledge base with specific objects - algorithms, methods, software tools, elements of parallel architectures, etc. Rules inside the SIS are set using the language SWRL (Sematic Web Rule Language) [12].

### 3. SIS for solving geophysical problems

This paragraph describes the authors' vision on presenting information in the form of an ontology for geophysical problems. This ontology is a multilevel structure that includes several ontologies as well. Note that ontologies are widely used in oil production [12, 13] geophysical studies [14] and others.

The knowledge base of SIS originates from the subject area - geophysics. For clarity, in fig. 2. the corresponding ontology is shown. Two main objects of research in this area are considered - a physical phenomenon and a physical object. A physical phenomenon is the excitation and propagation of wave fields, a physical object is the medium itself. Those two objects can be studied both through a field experiment using field observations and taking samples of the medium, and through a numerical experiment via mathematical modeling. A physical phenomenon is described using an approximate mathematical model. The choice of model depends on the statement of the problem and the object of interest. The obtained equations (one equation or a system of equations) with given initial and boundary conditions are then resolved using the numerical method. The numerical method is implemented using a parallel algorithm determined by a number of factors: what is the decomposition of the computational domain, what architecture is used for calculations. For a parallel algorithm, an optimized program code is constructed for performing calculations on parallel architectures. The calculations involve the necessary libraries and software components.

The upper level of ontologies of computational methods and parallel algorithms, as well as the ontology of parallel architectures are described in detail in [6, 7]. In the following figures 2 and 3, there are some components that show the link between elements of different ontologies.

#### 3.1 Ontology for solving a problem of wave propagation modeling

Lets analyze the ontological description in detail (Fig. 2) using as example the problem of modeling 3D wave fields in regions with complex surface topography [16]. The subject of the study is a physical phenomenon, namely the propagation of wave fields. In our case it is studied using a numerical experiment. The structure and properties of the medium are considered known. The medium is isotropic, the region of interest has a curved surface relief. The physical process is described by a linear elastic system. The system consists of 3 equations written in displacements, since the structure of the medium is assumed without reservoirs of liquid. This allows to significantly save RAM in further calculations on parallel architectures. Decomposition of the computational domain is represented by curved quadrangles, which allows good consistency between the boundaries of the numerical domain and the physical one. More details can be found in [16].

For numerical implementation, a numerical method is used, namely, an explicit finite-difference scheme. Using it, a parallel algorithm is created that is optimized for the case of three-dimensional decomposition of the computational domain. This method of partitioning the area is due to the large amount of data and is the most optimal. According to the algorithm, a parallel code is created that is executed on computers (in our case, Intel Xeon Phi accelerators are used, which are part of the SSCC SB RAS cluster). Code optimization is performed using AVX technology for vectorization and MPI package for organizing exchanges between computers, as well as parallelization inside each computer. This algorithm and its software implementation were investigated using simulation tools [10]. Note that numerical modeling is used not only to study the physics of seismic wave propagation, but also to test procedures and graphs for seismic data processing.

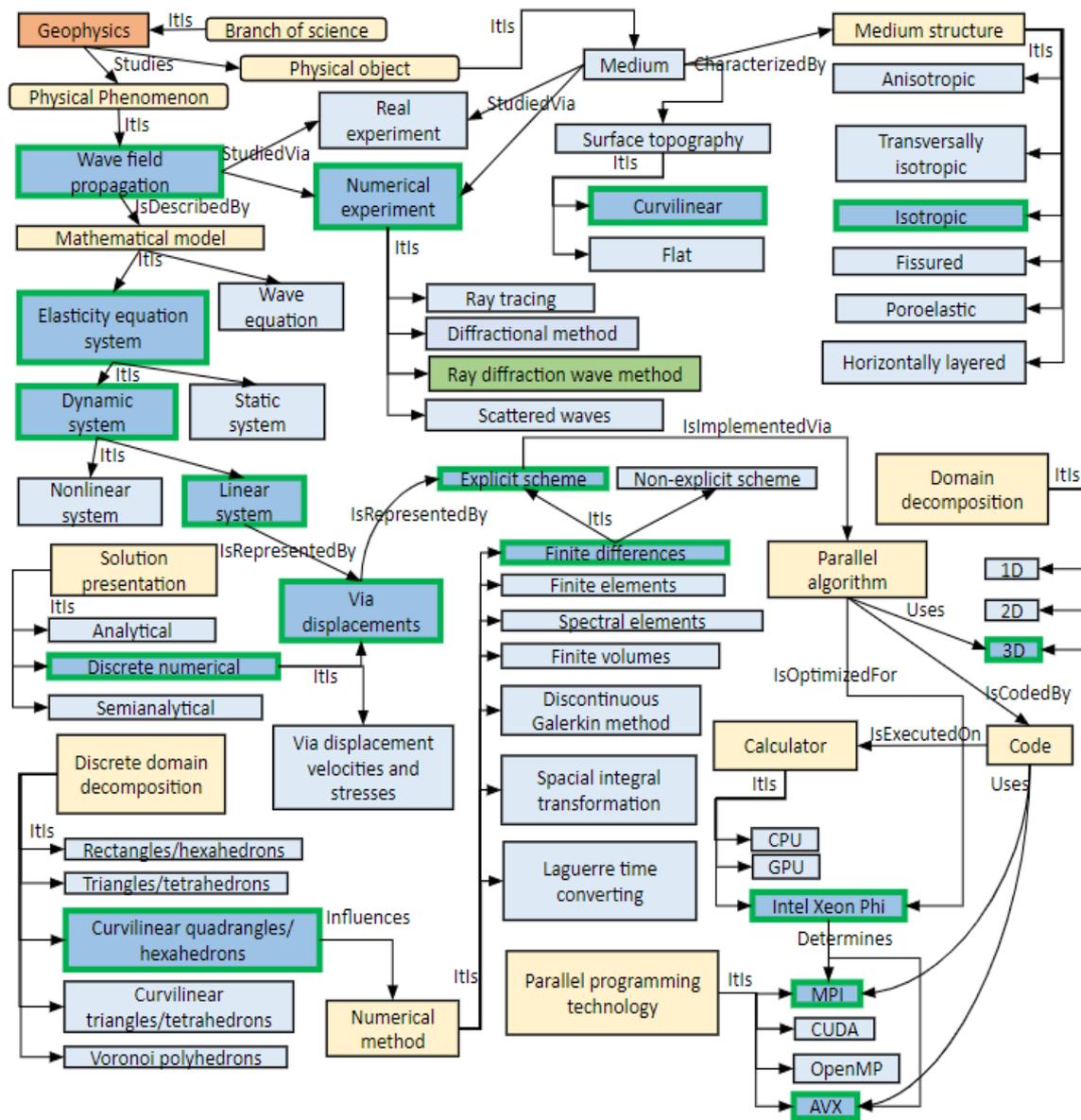


Fig. 2. Geophysical ontology. Blocks with thick frame are utilized in solving the problem of seismic field modeling [16].

### 3.2 Ontology for the problem of processing experimental seismic acquisition

Figure 3 shows the ontology used in the processing of field observation data to reconstruct dynamic images of the medium (seismogeological section) through the ray-diffraction model of the wave field. The basis is the restoration of averaged velocity models of the medium (as a rule, via P-waves) and the construction of stacked seismic profile from them. It is done using the seismic processing graph, which includes many components with feedback, the main of which are displayed in the diagram. Several components of the graph can be used simultaneously, all of them have a parallel implementation. Here we describe only a parallel implementation for the DMO-processing and 3D wave migration of a set of common offset section (cubes), using the stacking velocity model. The DMO processing is needed for an adequate implementation of the F-K migration scheme for each offset. A parallel algorithm is created, for which a parallel code is encoded, optimized for calculations on the CPU. Due to the small (<100GB) demand of the RAM, it is sufficient to use one computing node of the cluster with sufficient memory (for example, a node with two eight-core Intel Xeon E5-2697A v4 processors and 128 GB of RAM, NKS-1P cluster of the SSCC SB RAS). For calculations on a single node, it is enough to use the OpenMP package as well as AVX technology for vectorizing the code. As a rule, field data is transmitted in

the SEG-Y format, which does not fully include seismic acquisition parameters. A full description of it is contained in three types of SPS files,

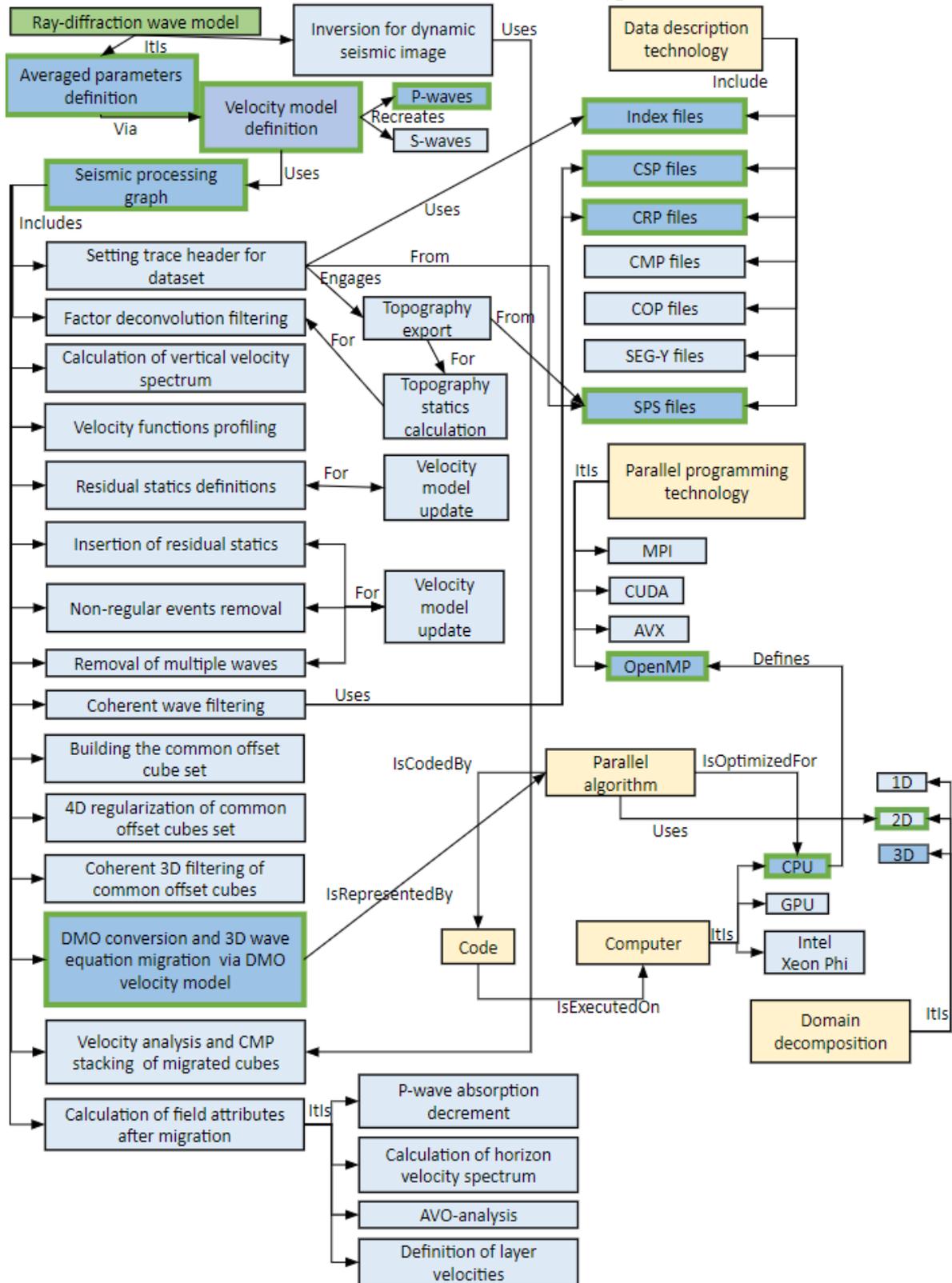


Fig. 3. Ontology of ray diffraction method with DMO-processing stage.

which are used to generate seismic trace headers in the required format. While processing, samples are required from the input data of seismic datasets via a defined key. Typically, these are all traces that are related to a common source point (CSP), receiving (CRP), common midpoint (CMP), and

total source-to-receiver offset. To select the required gathers, index files are created, consisting of trace numbers in the input dataset for each selection number (gathers). In general, the input data can be distributed over several files. Essentially, index files are a simplified multi-key database of dataset. A similar technology is used in almost all large packages of geophysical processing and interpretation, but is not described in publicly available sources [17], [18]. The figure 4 and the figure 5 show two vertical sections of a 3D stacked cube at the final stage of processing (in-line and cross-line) for a given region of Western Siberia. In-line sections are vertical 3D cube sections along CMP grid (along X axis), cross-line sections are vertical 3D cube sections across CMP grid (along Y axis). Sections of summarized cube (final 3D image of the medium under study) are obtained after velocity analysis of results of prestack time migration.

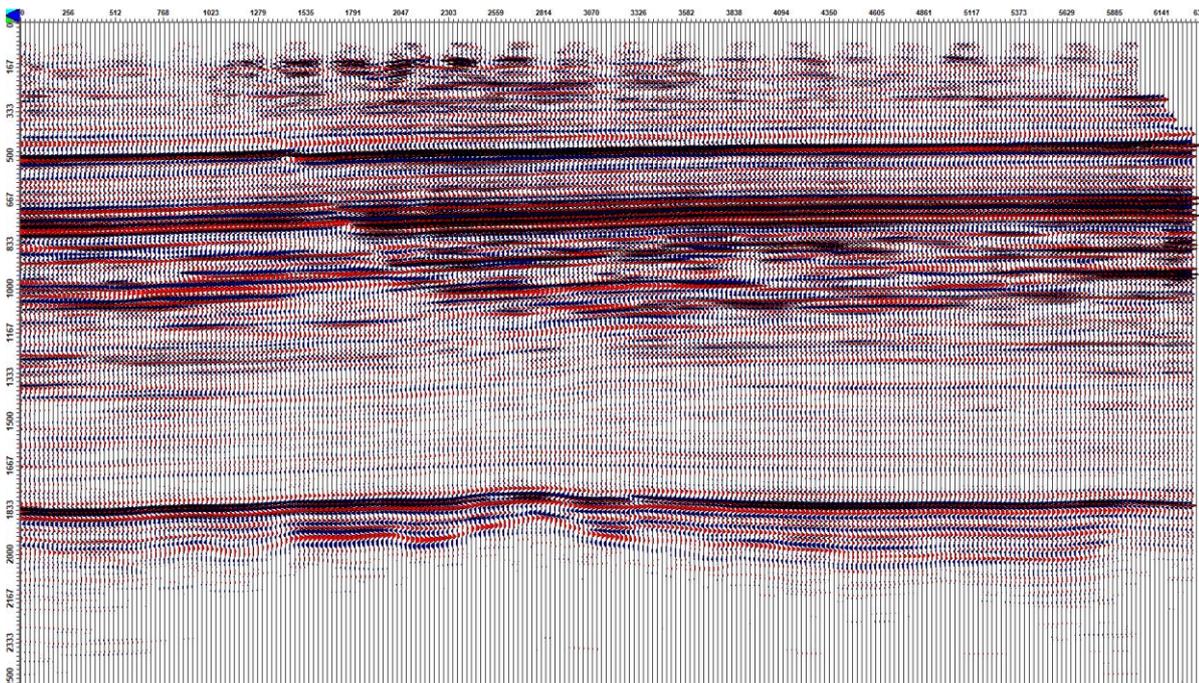
## 4. Conclusion

The paper presents the concept of an intelligent support system for solving compute-intensive problems of geophysics. A vision of the authors on the ontological concept of information in this subject area is presented. SIS (system of intelligent support) for solving geophysical problems combines methods and algorithms, available software packages as well as available supercomputer equipment. Thanks to the system, the user can significantly reduce the time spent to improve his qualifications as a specialist in this subject area, find the most suitable way to solve his problem, depending on requirements and preferences. Two examples are presented via means of presented SIS: 1) 3D wave field modeling in areas with topography 2) reconstructing a 3D seismic section for one of the regions of Western Siberia. Within the framework of the project, an Internet resource (<https://uniserv.iis.nsk.su/spsc>) was created, which will store the existing developments in the field of ontological representation of data in various subject areas. The resource will be systematically updated with new information as it becomes available.

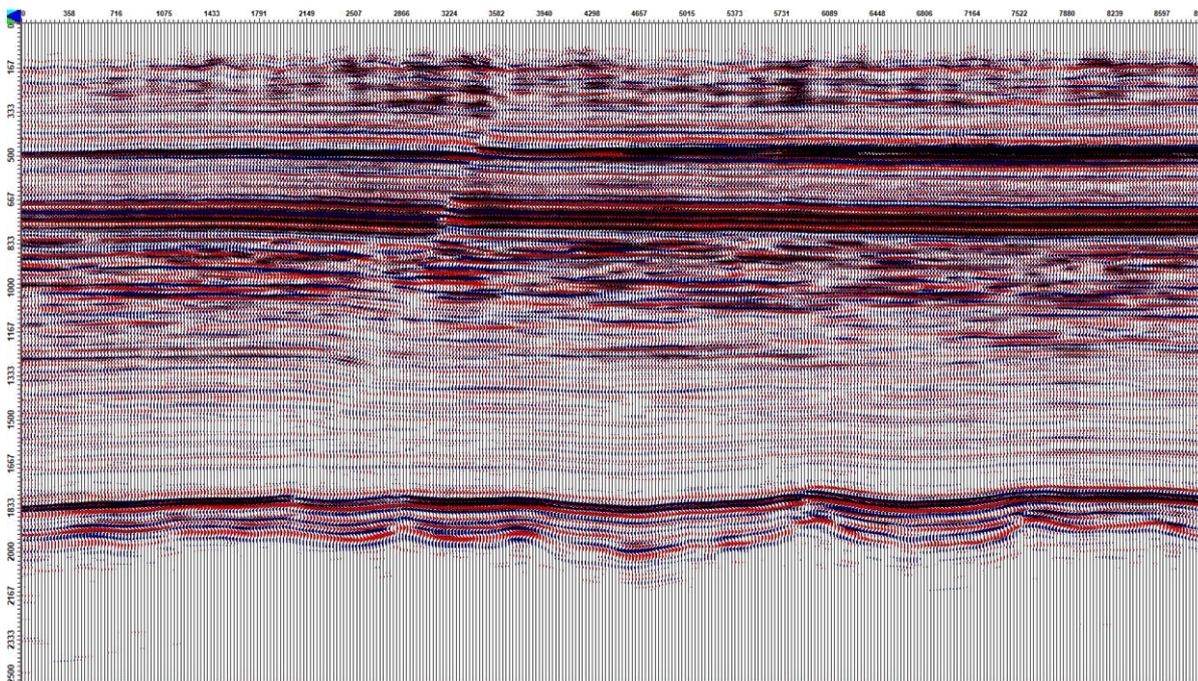
## Acknowledgements

This research was supported by budget project 0315-2019-0009 (sections 1), by the Russian Foundation for Basic Research [grants No. 19-07-00085] (sections 2, 3, 3.2), by the Russian Foundation for Basic Research [grants No. 20-01-00231] (section 3.1).

All numerical tests were conducted on SSCC ICMMG SB RAS.



**Fig. 4.** In-line section of a 3D stacked cube at the final stage of processing. Western Siberia.



**Fig. 5.** Cross-line section of a 3D stacked cube at the final stage of processing. Western Siberia.

## Literature

1. Compton, M.; Barnaghi, P.; Bermudez, L.; Garcia-Castro, R.; Corcho, O.; Cox, S.; Graybeal, J.; Hauswirth, M.; Henson, C.; Herzog, A.; et al. The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Web Semant. Sci. Serv. Agents World Wide Web 2012*, 17, 25–32.
2. Keet, C.M.; Ławrynowicz, A.; d’Amato, C.; Kalousis, A.; Nguyen, P.; Palma, R.; Stevens, R.; Hilario, M.H. The Data Mining OPTimization Ontology. *Web Semant. Sci. Serv. Agents World Wide Web 2015*, 32, 43–53.
3. Ma, X.: *Ontology Spectrum for Geological Data Interoperability*. ITC, Netherlands (2011).
4. Cvjetkovic, V.: Web physics ontology: online interactive symbolic computation in physics. In: 2017 4th Experiment@International Conference (Exp.at’17), Faro, pp. 52–57 (2017).
5. Sarro, L.M., Martínez, R.: First steps towards an ontology for astrophysics. In: Palade, V., Howlett, R.J., Jain, L. (eds.) *KES 2003. LNCS (LNAI)*, vol. 2774, pp. 1389–1395. Springer, Heidelberg (2003). [https://doi.org/10.1007/978-3-540-45226-3\\_188](https://doi.org/10.1007/978-3-540-45226-3_188).
6. B. Glinskiy, Y. Zagorulko, G. Zagorulko, I. Kulikov, A. Sapetina. The Creation of Intelligent Support Methods for Solving Mathematical Physics Problems on Supercomputers. *Russian Supercomputing Days 2019*, Springer International Publishing 2019, 427-438, DOI 10.1007/978-3-030-36592-9\_35.
7. G. Zagorulko, Y. Zagorulko, B. Glinskiy, A. Sapetina. (2019) Ontological Approach to Providing Intelligent Support for Solving Compute-Intensive Problems on Supercomputers. In: Kuznetsov S., Panov A. (eds) *Artificial Intelligence. RCAI 2019. Communications in Computer and Information Science*, vol 1093, pp 363-375.
8. Cook, D., Neal, M., Bookstein, F., Gennari, J.: Ontology of physics for biology: representing physical dependencies as a basis for biological processes. *J Biomed. Semant.* 4, 41 (2013).

9. Glinskiy, B., Sapetina, A., Martynov, V., Weins, D., Chernykh, I. The Hybrid-Cluster Multilevel Approach to Solving the Elastic Wave Propagation Problem (2017) Communications in Computer and Information Science, 753, pp. 261-274. DOI: 10.1007/978-3-319-67035-5\_19.
10. P.A. Titov, D.V. Weins, I.G. Chernykh. Application of an integral approach to the parallel algorithm of 3D wave fields simulation in generalized coordinates // Russian supercomputing days: workshop, sep 23-24 2019 г., Moscow Russia – Moskva : MAKS Press, 2019, pp. 14-25.
11. Antoniou, G., Harmelen, F.: Web Ontology Language: OWL. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 67-92. Springer Verlag, Berlin (2004).
12. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. <http://www.w3.org/Submission/SWRL/> , SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. <http://www.w3.org/Submission/SWRL/>.
13. Didyk T.G., Rykov V.I., Sharonova Yu.V. ALGORITHMS AND MEANS OF FORMING ONTOLOGY OF A TASKED SUBJECT AREA // Modern problems of science and education. - 2013. - No. 6 (In Russian).
14. Belaid N., Ait-Ameur Y. и Rainaud J. (2009). SWAPT - semantic workflow architecture for petroleum technology. *WEBIST, vol.1* ISBN 978-989-8111-81-4, стр. 101-104. DOI: 10.5220 / 0001840101010104.
15. Viral Parekh, Jin-Ping Gwo, Tim Finin. Ontology based on semantic metadata for geophysical research. (Translation in russian by A. Bazhanov) <http://masters.donntu.org/2011/fknt/bazhanova/library/>.
16. P.A. Titov. The simulation of 3D wave fields in complex topography media // 5th Russian Supercomputing Days, RuSCDays 2019, Moscow, Russia, September 23–24, 2019, Revised Selected Papers, Communications in Computer and Information Science, 451-462, DOI 10.1007/978-3-030-36592-9\_37.
17. Liseykin, V.: Difference Meshes. Theory and Applications, p. 254. SB RAS Publishing, Novosibirsk (2014) (in Russian).
18. R.E. Sheriff, L.P. Geldart. Exploration Seismology. Cambridge University Press, 1995.
19. Yilmaz O. Seismic data processing. Tulsa: SEG, 1986.

# Архитектура, характеристики и функциональные возможности коммуникационной сети Ангара второго поколения<sup>1</sup>

А.С. Симонов, А.С. Семенов, Д.В. Макагон

АО «Научно-исследовательский центр электронной вычислительной техники»

В статье обобщён опыт оценочного тестирования и практической эксплуатации многопроцессорных вычислительных систем, построенных на основе коммуникационной сети Ангара первого поколения, описаны основные архитектурные решения коммуникационной сети Ангара второго поколения и представлен прогноз её характеристик, полученный по результатам имитационного моделирования и макетирования с использованием ПЛИС.

*Ключевые слова:* коммуникационная сеть Ангара, многопроцессорная вычислительная система, суперкомпьютер

## 1. Введение

После появления NEC SX-6 Earth Simulator [1] стало понятно, насколько важны характеристики и возможности межузловой коммуникационной сети для обеспечения высокой масштабируемости производительности вычислительных кластеров и суперкомпьютеров (далее многопроцессорных вычислительных систем, МВС) при решении вычислительно сложных задач, прежде всего задач компьютерного моделирования, обработки больших массивов данных, планирования и прогнозирования.

Разработка высокоскоростной коммуникационной сети началась в АО «НИЦЭВТ» в 2006 году. Анализ мирового опыта по созданию заказных коммуникационных сетей для суперкомпьютеров высшего диапазона производительности, прежде всего IBM серии BlueGene [2-4] и CRAY SeaStar/Gemini [5-7], исследования, проведённые в АО «НИЦЭВТ» с использованием средств имитационного моделирования [8], позволили сформировать принципы работы и технический облик высокоскоростной коммуникационной сети Ангара первого поколения (далее ВКС Ангара G1) [9], серийные образцы которой были представлены в 2013 году. ВКС Ангара G1 представляет собой сеть класса Direct Network, поддерживает топологии от 1D-mesh до 4D-тор и обеспечивает возможность создания МВС размером до 32 тысяч узлов.

Опыт, полученный при создании Ангара G1 и МВС на её основе, позволил разработать архитектурные решения и сформировать технический облик коммуникационной сети Ангара второго поколения (далее ВКС Ангара G2), разработка ключевого компонента которой – заказной СБИС – в настоящее время находится на завершающей стадии.

В данной работе подробно описано первое поколение сети Ангара, представлены основные архитектурные решения высокоскоростной коммуникационной сети Ангара второго поколения, приведён полученный по результатам имитационного моделирования и макетирования прогноз её характеристик.

## 2. Коммуникационная сеть Ангара первого поколения

Целью разработка высокоскоростной сети Ангара являлось обеспечение возможности создания высокопроизводительных вычислительных систем, обладающих свойством высокой масштабируемости производительности на широком диапазоне прикладных задач.

---

<sup>1</sup> Работы по созданию коммуникационной сети Ангара проводятся при поддержке Минпромторга РФ.

## 2.1 Архитектура и варианты исполнения сетевого оборудования Ангара первого поколения

Один из самых важных вопросов при разработке архитектуры высокоскоростной сети – выбор топологии. Среди множества известных топологий [10], среди которых в первую очередь необходимо назвать  $n$ -мерный тор и топологию «жирное дерево», в ходе разработки ВКС Ангара G1 был сделан выбор в пользу тороидальной топологии, при этом ниже сформулированы основные аргументы в пользу данного решения. Во-первых, топология  $n$ -мерный тор получила широкое распространение среди заказных зарубежных МВС высшего диапазона производительности, выпущенных ведущими зарубежными компаниями: суперкомпьютеры IBM серии Blue Gene /L/P/Q, CRAY серии XT/XE/XK, Fujitsu K computer [11]. Во-вторых, при реализации данной топологии технические риски оценивались как более низкие по сравнению с другими топологиями, а также существовала технологическая возможность создания законченного серийного коммуникационного оборудования. Кроме того, на высоком уровне находилась теоретическая и практическая отработка существующих алгоритмов маршрутизации [12], которые позволяли обеспечить устойчивость к отказам отдельных элементов. Было проведено имитационное моделирование [8], которое показало высокую толерантность топологии  $n$ -мерный тор к всплескам нагрузки и достаточные характеристики производительности при выполнении глобальных паттернов сетевых обменов типа «все со всеми». Также очевидна высокая эффективность данной топологии при решении задач с использованием сеточных методов.

Выбор топологии определил круг применимых алгоритмов маршрутизации и методов решения основных проблем коммуникационных сетей – возникновения взаимных блокировок, голодания и закливаний. В результате комплекса исследований были обоснованно приняты следующие решения по архитектуре ВКС Ангара G1:

- топология  $n$ -мерный тор;
- детерминированный алгоритм маршрутизации с фиксированным порядком направлений (англ. directional order routing) [10] с возможностью первого и последнего шагов в произвольном направлении для обхода отказавших узлов [13];
- минимальный полноадаптивный алгоритм маршрутизации с заданием возможных направлений;
- виртуальные подсети для детерминированной, адаптивной и широковещательной маршрутизации на основе виртуальных каналов;
- дисциплина передачи данных Virtual cut-through (VCT);
- защита от взаимной блокировки (англ. deadlock) в детерминированных виртуальных каналах методом «пузырька» [14], защита от взаимной блокировки в адаптивной подсети путём перенаправления пакета в детерминированную подсеть по таймауту;
- защита от закливания (англ. livelock) за счёт использования алгоритмов маршрутизации, в которых каждый шаг движения пакета сокращает расстояние до узла назначения;
- защита от голодания (англ. starvation) за счёт использования справедливых алгоритмов арбитража;
- интерфейс с процессором вычислительного узла PCI Express;
- аппаратная поддержка технологии RDMA прямой записи в память удаленного узла;
- Onload и Offload варианты исполнения для бюджетного и высокопроизводительного сегментов применения соответственно;
- аппаратно-программная поддержка основных функций библиотек SHMEM и MPI;
- функциональная реализация маршрутизатора в виде заказной СБИС.

На основе заказной СБИС были созданы варианты исполнения сетевого оборудования ВКС Ангара G1:

- в форм-факторе полноформатного адаптера PCI Express (см. рис. 1), обеспечивающего возможность непосредственного объединения до 32 тысяч вычислительных узлов МВС в топологию до 4D-тор со сторонами  $8 \times 16 \times 16 \times 16$ ;

- в форм-факторе 24-портового коммутатора для монтажа в стандартную телекоммуникационную стойку и low profile адаптера PCI Express для установки в вычислительные узлы МВС (см. рис. 2). Данный вариант обеспечивает возможность объединения до 2048 вычислительных узлов путём соединения в топологию 2D-тор до 256 коммутаторов.



Рис. 1. ВКС Ангара G1 в форм-факторе полноформатного адаптера PCI Express



Рис. 2. ВКС Ангара G1 в форм-факторе 24-портового коммутатора (а) и low profile адаптера PCI Express (б)

## 2.2 Опыт создания и эксплуатации многопроцессорных вычислительных систем, построенных на основе коммуникационной сети Ангара первого поколения

Первой вычислительной системой, оборудованной ВКС Ангара G1, стал 36-узловой вычислительный кластер «Ангара-K1» [15], установленный в АО «НИЦЭВТ» и предназначенный для практической отработки аппаратного и системного программного обеспечения коммуникационной сети при решении прикладных задач.

В ходе отработки были проверены различные варианты объединения вычислительных узлов в топологию n-мерный тор, проведена отладка системного программного обеспечения (см. рис. 3), планировщика задач Slurm [16], плагинов системы мониторинга Zabbix.



**Рис. 3.** Системное программное обеспечение ВКС Ангара G1

Для проведения оценочного тестирования ВКС Ангара G1 к вычислительному кластеру «Ангара-K1» предоставлен удаленный доступ, и за прошедшее время тестовые расчёты были проведены несколькими десятками организаций. Результатом этой работы стало создание ряда суперкомпьютеров различной производительности на основе ВКС Ангара G1, в том числе суперкомпьютеров «Десмос» [17] (рис. 4) и «Фишер» [18], установленных в ОИВТ РАН.

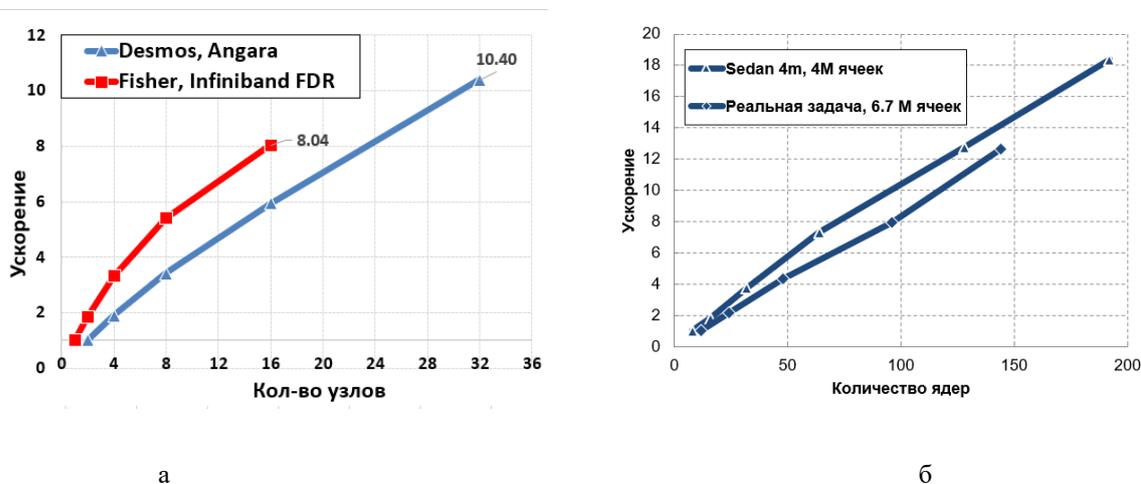


**Рис. 4.** Суперкомпьютер «Десмос» на основе ВКС Ангара G1

Очевидно, что при создании и эксплуатации указанных МВС были выявлены как преимущества, так и недостатки ВКС Ангара G1. К преимуществам относятся:

- крайне низкая коммуникационная задержка, а именно – 850 нс с использованием библиотеки MPI на тесте `osu_latency`, имеющая хорошую, по сравнению с другими коммерчески доступными коммуникационными сетями, устойчивость с ростом нагрузки на вычислительную сеть и при увеличении числа вычислительных узлов МВС;
- высокая реальная пропускная способность, не изменяющаяся с увеличением расстояния между узлами. Данное свойство обеспечивается как выбранной топологией, так и хорошей сбалансированностью стадий всего конвейера передачи данных ВКС Ангара G1.

Перечисленные достоинства позволили получить хорошие показатели масштабируемости производительности МВС, построенных на основе ВКС Ангара G1, как на оценочных тестах, так и при решении прикладных задач компьютерного моделирования (см. рис. 5) [17-22].



**Рис. 5.** Результаты масштабируемости производительности МВС (а – пакет программ для проведения квантово-химических расчетов VASP, суперкомпьютер «Десмос»; б – пакет ANSYS FLUENT, вычислительный кластер «Ангара-К1»)

Однако были выявлены и недостатки ВКС Ангара G1. К ним, в первую очередь, относятся:

- некоторое ухудшение основных характеристик МВС при выходе из строя отдельных вычислительных узлов МВС или кабелей связи, связанное с ограниченными возможностями применённых алгоритмов маршрутизации при обходе отказавших участков сети;
- некоторое ухудшение основных характеристик МВС при перегрузке отдельных каналов или участков сети, также связанное с особенностями выбранного адаптивного алгоритма маршрутизации;
- недостаточная аппаратная поддержка отказоустойчивости в заказной СБИС, включая поддержку сохранения и восстановления состояния параллельно выполняющихся задач из контрольных точек;
- недостаточная аппаратная поддержка современных многоядерных процессоров;
- недостаточная аппаратная поддержка интеграции с GPU;
- отсутствие аппаратной поддержки стека протоколов TCP/IP;
- отсутствие аппаратной поддержки виртуализации;
- недостаточная аппаратная поддержка систем мониторинга и управления МВС, что особенно важно для крупных МВС высшего диапазона производительности;
- большие габаритные размеры ВКС Ангара G1 в форм-факторе полноформатного адаптера PCI Express;
- ограничение по числу узлов МВС при использовании ВКС Ангара G1 в форм-факторе 24-портового коммутатора и low profile адаптера PCI Express.

Перечисленные недостатки были учтены при разработке ВКС Ангара G2.

## 2. Коммуникационная сеть Ангара второго поколения

Высокоскоростная сеть Ангара второго поколения предназначена для объединения вычислительных узлов МВС вплоть до субэкзэфлопсного уровня производительности. При её разработке соблюдались принципы эволюционного развития, то есть минимизация технических рисков, учет полученного опыта при разработке и эксплуатации сети Ангара первого поколения и устранение выявленных недостатков. Кроме сохранения цели по обеспечению высокой масштабируемости производительности построенных с использованием разрабатываемой сети суперкомпьютеров, была поставлена цель расширения спектра применения коммуникационной сети, которую предполагается использовать не только в суперкомпьютерах, но и в системах хранения данных, системах обработки Больших Данных и в центрах обработки данных.

Сеть Ангара G2 предполагается использовать в составе вычислительных систем произвольного размера, однако отдельное внимание разработчиков было уделено обеспечению соответствия сети масштабу МВС среднего и высшего диапазона производительности на перспективу 4-5 лет. В связи с этим ставилась задача поддержания показателя сбалансированности между производительностью вычислительных узлов МВС и пропускной способностью коммуникационной сети на уровне не хуже 100 Гфлопс на Гбайт/с. Для достижения необходимой производительности, а также вариативности применения при создании различных вариантов исполнения сетевого оборудования основу коммуникационной сети должны составлять заказные СБИС.

Также для обеспечения масштабируемости производительности приложений и поддержки новых областей применения было необходимо:

- использование более гибких алгоритмов маршрутизации;
- низкая коммуникационная задержка;
- повышение темпа выдачи сообщений;
- расширение функциональных возможностей коммуникационной сети.

## 2.1 Архитектура и варианты исполнения сетевого оборудования Ангара второго поколения

Разработанная с учетом поставленных целей архитектура ВКС Ангара G2 включает следующие основные отличия по сравнению с Ангара G1:

- гибкие варианты комбинирования доступных топологий от 1D-решетки до 6D-тор, поддерживаются также топологии со скошенными связями;
- для обеспечения сбалансированности пропускная способность на вычислительный узел может выбираться из ряда 100 Гбит/с, 200 Гбит/с или 400 Гбит/с;
- разработаны алгоритмы маршрутизации *delta routing*, обеспечивающие более гибкие варианты маршрутизации при обходе отказавших или перегруженных участков сети;
- разработан алгоритм зональной адаптивной маршрутизации;
- поддерживаются низкоуровневые сетевые операции с подтверждениями и уведомлениями, операции синхронизации,
- добавлена аппаратная поддержка *multihost*, позволяющая обеспечить одновременное использование в каждом вычислительном узле 2 интерфейсов PCI Express 4.0 x16;
- улучшена поддержка современных многоядерных процессоров, аппаратно реализованы 128 независимых инжекционных конвейеров;
- улучшены аппаратные способы преодоления проблемы голодания, то есть несправедливого распределения ресурсов;
- существенно расширена поддержка отказоустойчивости, а также систем мониторинга и управления МВС. С этой целью в архитектуру введены поддержка дополнительных исключительных ситуаций, значительно расширен состав и гибкость аппаратных средств мониторинга и управления заказной СБИС.
- реализована аппаратная поддержка виртуализации SR-IOV, в каждом PCI Express endpoint поддерживаются до 15 виртуальных функций;
- улучшена аппаратная поддержка горячей замены вышедшего из строя сетевого оборудования.

Важнейшим нововведением в Ангара G2 по сравнению с сетью Ангара G1 является переход к поддержке топологии до 6-мерного тора. Данное изменение архитектуры следует мировой тенденции развития коммуникационных сетей по увеличению связности каждого узла или коммутатора.

Алгоритм маршрутизации *delta routing* реализует правило маршрутизации *dimension order routing*, один из вариантов которого (англ. *direction order routing*) был использован при разработке ВКС Ангара G1. Алгоритм *dirbit*-маршрутизации, примененный в Ангара G1, подразумевает хранение в заголовке пакета не только адреса целевого узла, но и битов направлений передачи. В алгоритме *delta routing* в заголовке пакета для каждого измерения необходимо хранить лишь разницу между текущей и целевой координатой. В отличие от *dirbit*-маршрутизации, алгоритм *delta routing* является более гибким и позволяет передавать пакет и в положительную, и в отрицательную сторону по измерению, причем на большое количество шагов, что позволяет существенно расширить возможности по обходу отказавших вычислительных узлов МВС или кабелей связи. Кроме того, алгоритм *delta routing* позволяет строить скошенные связи в торе, что в некоторых случаях может сократить число шагов и улучшить показатели коммуникационной задержки.

Другим важным усовершенствованием является возможность зональной адаптивной маршрутизации. В ВКС Ангара G1 при движении пакета с использованием адаптивной маршрутизации в случае возникновения каких-либо проблем, например, слишком долгого ожидания пакетом соединения, происходил переход пакета в детерминированную подсеть, причем обратно к использованию адаптивной маршрутизации перейти было нельзя. Зональная адаптивная маршрутизация позволяет задать, в каких направлениях пакет может передаваться с использованием адаптивного алгоритма, а в каких – с использованием детерминированного. Таким об-

разом, передача пакетов с использованием детерминированного алгоритма будет использоваться лишь в гиперплоскостях, в которых имеются какие-либо проблемы, например, отказы вычислительных узлов или каналов связи между ними.

В ВКС Ангара G2 значительно расширена номенклатура низкоуровневых операций, помимо операций put, get и атомарных операций (в т.ч. с возвратом значения) в памяти удалённого узла аппаратно поддерживаются операции записи в сегменты (доступно 3 вида сегментов – подсчёта пришедших пакетов, агрегирующий, кольцевой), операции с подтверждениями, уведомлениями (в т.ч. с использованием прерываний). Использование данных типов операций позволяет оптимизировать реализации низкоуровневого программного обеспечения библиотек MPI и TCP/IP, снизить нагрузку на центральный процессор и существенно улучшить производительность и масштабируемость приложений. Важно отметить, что в Ангара G2 по-прежнему поддерживается технология прямой записи в память удаленного узла (RDMA), которая значительно влияет на достигаемую производительность не только в сегменте высокопроизводительных систем, но и в системах хранения данных и обработки Больших Данных.

Архитектура СБИС маршрутизатора ВКС Ангара G2 представлена на рис. 6 и состоит из двух частей: собственно маршрутизатора и адаптера. Маршрутизатор предназначен для маршрутизации и передачи пакетов между узлами, включает четыре виртуальных канала в каждом линке, образующих: три подсети детерминированной передачи для запросов, ответов, подтверждений и подсеть адаптивной передачи. Виртуальные каналы детерминированной передачи позволяют реализовать детерминированный алгоритм маршрутизации с перебором направлений в порядке, который можно задавать при старте системы.

Адаптер выполняет инъекцию пакетов в сети и обработку пакетов, эжектируемых из сети, а также набор сервисных функций, обеспечивающих минимизацию трафика по сети и через интерфейс с процессором узла. В его состав входят:

- блоки управления интерфейсами, поддерживающие работу по протоколам PCI Express и реализующие поддержку виртуальных функций;
- блоки инъекции и эжекции, формирующие пакеты на отправку в сеть и разбирающие заголовки пакетов, пришедших из сети;
- блок служебных операций, обрабатывающий пакеты, идущие в сервисный процессор и из него;
- блоки агрегации и синхронизации, а также блок подтверждений и уведомлений, выполняющие поддержку выполнения различных сетевых операций;
- блоки неблокирующих и атомарных операций, а также операций чтения, которые реализуют поддержку технологии RDMA.

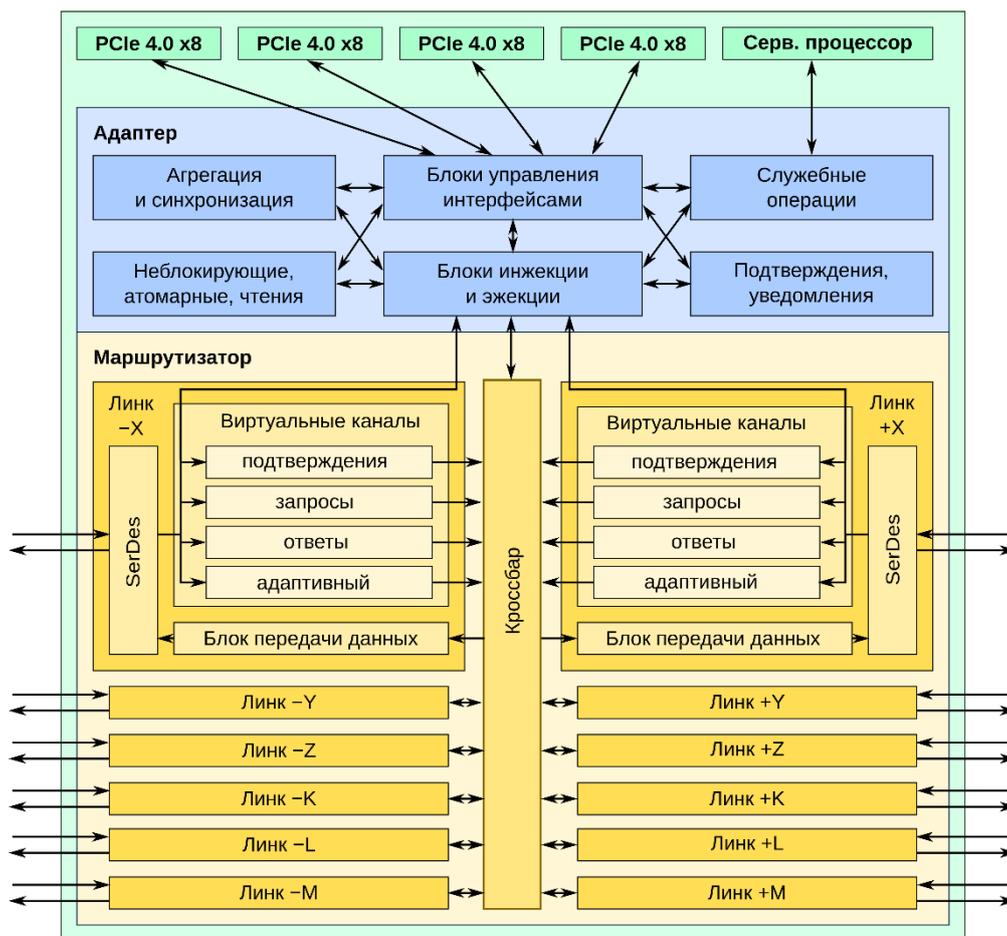


Рис. 6. Архитектура СБИС маршрутизатора ВКС Ангара G1

Для представления более ясного облика сети Ангара G2 важен вид планируемого сетевого оборудования, которое планируется выпускать в следующих вариантах исполнения:

- в форм-факторе полноформатного адаптера PCI Express, обеспечивающего возможность непосредственного объединения до 1024 вычислительных узлов МВС в топологию до 3D-тор со сторонами 16x8x8. Адаптер будет включать 6 портов с разъёмами QSFP-DD по 200 Гбит/с каждый и поддерживать работу в режиме multihost до двух интерфейсов PCI Express 4.0 x16.
- в форм-факторе 40-портового коммутатора для монтажа в стандартную телекоммуникационную стойку и low profile адаптера PCI Express для установки в вычислительные узлы МВС. Данный вариант обеспечивает возможность объединения до 20480 вычислительных узлов путём соединения коммутаторов в топологию 4D-тор. Каждый коммутатор будет включать по 40 портов с разъёмами QSFP-DD 200 Гбит/с, а low profile адаптер – по 2 порта с разъёмами QSFP-DD 200 Гбит/с и поддерживать работу в режиме multihost до двух интерфейсов PCI Express 4.0 x16.
- заказное оборудование, разработанное с учетом требований заказчика с возможностью построения сети с топологией до 6D-тор и пропускными способностями каналов связи 100 Гбит/с, 200 Гбит/с или 400 Гбит/с.

## 2.2 Прогноз технических характеристик многопроцессорных вычислительных систем, построенных на основе сетевого оборудования Ангара второго поколения

Отработка заложенных архитектурных решений сети Ангара G2 осуществлялась на имитационной модели, а также на 9-узловом кластере с макетными образцами сетевых адаптеров на базе ПЛИС Virtex7 vx690tffg, в составе каждого вычислительного узла 2 процессора Intel Xeon CPU E5-2630 v3 @ 2.40 ГГц, 32 ГБ памяти.

Макетные образцы выполнены в форм-факторе полноформатной платы расширения PCI Express. Из-за технических ограничений ПЛИС в качестве интерфейса с процессором использовался PCI Express 2.0 x8, инъекция пакетов осуществлялась через 4 независимых инъекционных конвейера, для связи с другими адаптерами использовались 8 разъёмов/кабелей QSFP.

На макете при рабочей частоте адаптера в 167 МГц удалось получить коммуникационную задержку между соседними узлами на тесте `osu_latency` с использованием библиотеки MPI – 2.01 мкс, график результатов выполнения теста приведен на рис. 7. Задержка на каждый дополнительный хоп составляет не более 0.5 мкс. Пропускная способность была ограничена скоростью инъекции пакетов в сеть – в пределах 25–30 Гбит/с, накладные расходы протокола передачи данных сети Ангара G2 при передаче больших сообщений составляют менее 9% от пиковой скорости канала связи.

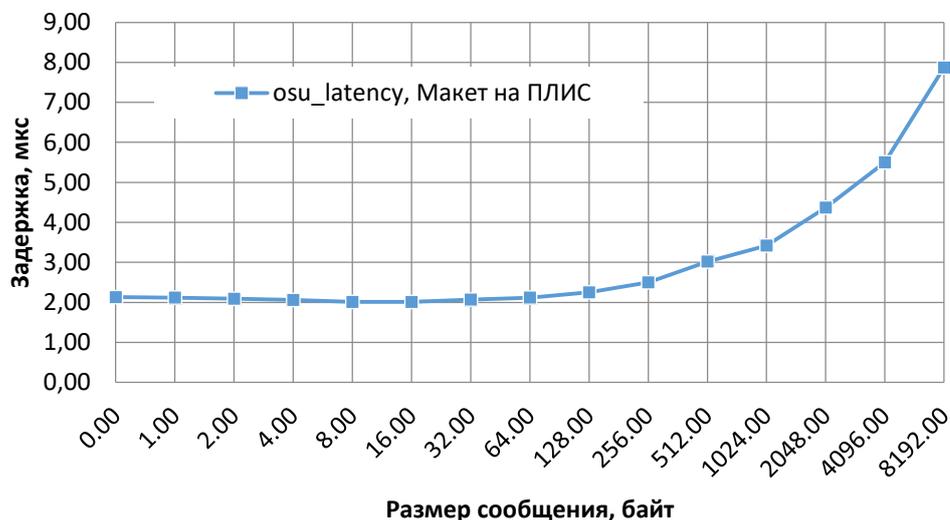


Рис. 7. Задержка передачи сообщения, полученная между двумя узлами на макете Ангара G2 на ПЛИС, тест `osu_latency`

Ожидается, что СБИС сети Ангара G2 будет работать на частоте 1.0–1.2 ГГц. Коммуникационная задержка между соседними узлами составит не более 1 мкс, задержка на каждый дополнительных хоп – менее 0.1 мкс. Чистая скорость инъекции пакетов в сеть при подключении через интерфейс PCI Express 4.0 x16 ожидается на уровне 180 Гбит/с, чистая достижимая пропускная способность канала связи при передаче больших сообщений – также на уровне 180 Гбит/с.

## 3. Выводы

В статье представлены основные технические решения по архитектуре высокоскоростной коммуникационной сети Ангара первого поколения, в том числе по топологии, алгоритмам адаптивной и детерминированной маршрутизации, технологии RDMA прямой записи в память удаленного узла. Представлены разработанные варианты сетевого оборудования и результаты оценки масштабируемости производительности суперкомпьютеров, построенных на основе сети Ангара первого поколения G1, при решении прикладных задач компьютерного моделирования.

На основе опыта оценочного тестирования и практической эксплуатации многопроцессорных вычислительных систем, построенных на основе коммуникационной сети Ангара первого поколения, представлены основные архитектурные решения коммуникационной сети Ангара второго поколения, включающие поддержку топологии до 6D-тор, пропускную способность каналов связи 100 Гбит/с, 200 Гбит/с и 400 Гбит/с, новые сетевые операции и алгоритмы маршрутизации, поддержку технологии виртуализации SR-IOV и другие возможности. Результаты макетирования сети Ангара второго поколения на ПЛИС подтверждают возможность получения целевых характеристик производительности для разрабатываемого сетевого оборудования на базе СБИС.

## Литература

1. Habata S. et al. The earth simulator system // NEC Research and Development. – 2003. – Т. 44. – №. 1. – С. 21–26.
2. Gara A. Overview of the Blue Gene/L system architecture // IBM Journal of research and development. – Vol. 49. – 2005. – Pp. 195–212. – [http://rsim.cs.illinois.edu/arch/qual\\_papers/systems/19.pdf](http://rsim.cs.illinois.edu/arch/qual_papers/systems/19.pdf) (дата обращения: 08.10.2018).
3. Overview of the IBM Blue Gene/P project / Gheorghe Almasi, Sameh Asaad, Ralph E Bellofatto et al. // IBM Journal of Research and Development. – 2008. – Vol. 52, no. 1–2. – Pp. 199–220. – <http://scc.acad.bg/ncsa/documentation/team.pdf> (дата обращения: 08.10.2018).
4. The ibm blue gene/q interconnection fabric / Dong Chen, Noel Eisley, Philip Heidelberger et al. // IEEE Micro. – 2012. – Vol. 32, no. 1. – Pp. 32–43. – [https://www.researchgate.net/publication/220290398\\_The\\_IBM\\_blue\\_geneQ\\_interconnection\\_fabric](https://www.researchgate.net/publication/220290398_The_IBM_blue_geneQ_interconnection_fabric) (дата обращения: 08.10.2018).
5. Abts D, Storm C R. The Cray XT4 and Seastar 3-D torus interconnect. – 2010. <https://static.googleusercontent.com/media/research.google.com/ru/pubs/archive/36896.pdf> (дата обращения: 08.10.2018).
6. Alam S R. Cray XT4: an early evaluation for petascale scientific simulation // Proceedings of the 2007 ACM/IEEE Conference on / IEEE. – 2007. – Pp. 1–12. – <https://doi.org/10.1145/1362622.1362675> (дата обращения: 09.10.2018).
7. Alverson R, Roweth D, Kaplan L. The gemini system interconnect // High Performance Interconnects (HOTI), 2010 IEEE 18th Annual Symposium / IEEE. – 2010. – Pp. 83–87. – <https://doi.org/10.1109/HOTI.2010.23> (дата обращения: 09.10.2018).
8. Симонов А. С. Имитационная модель высокоскоростной коммуникационной сети Ангара с топологией kD-тор // Труды МАИ. – 2019. – №. 109. – С. 22–22.
9. Жабин И. А. и др. Первое поколение высокоскоростной коммуникационной сети "Ангара" // Наукоемкие технологии. – 2014. – Т. 15. – №. 1. – С. 021–027.
10. Duato J., Yalamanchili S., Ni L. Interconnection networks. – Morgan Kaufmann, 2003.
11. Ajima Y. et al. Tofu: Interconnect for the K computer // Fujitsu Sci. Tech. J. – 2012. – Т. 48. – №. 3. – С. 280–285.
12. Dally W. J., Towles B. P. Principles and practices of interconnection networks. – Elsevier, 2004.
13. Scott Steven L et al. The Cray T3E network: adaptive routing in a high performance 3D torus. – 1996. <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=7B4A3FAE3AC52B4E4141206064C00923?doi=10.1.1.126.3882&rep=rep1&type=pdf> (дата обращения: 08.10.2018).
14. Adaptive bubble router: a design to improve performance in torus networks /Valentin Puente, Ram'on Bevide, Jos'e A Gregorio et al. // Parallel Processing, 1999. Proceedings. 1999 International Conference on / IEEE. – 1999. – Pp. 58–67. – <https://doi.org/10.1109/ICPP.1999.797388> (дата обращения: 09.10.2018).
15. Исмагилов Т. Ф., Семенов А. С., Симонов А. С. Результаты оценочного тестирования отечественной высокоскоростной коммуникационной сети Ангара // Суперкомпьютерные дни в России. – 2016. – С. 626–639.
16. Mukosey A., Simonov A., Semenov A. Extended Routing Table Generation Algorithm for the Angara Interconnect // Russian Supercomputing Days. – Springer, Cham, 2019. – С. 573–583.

17. Stegailov, V., Dlinnova, E., Ismagilov, T., Khalilov, M., Kondratyuk, N., Makagon, D., Semenov, A., Simonov, A., Smirnov, G., Timofeev, A.: Angara interconnect makes GPU-based Desmos supercomputer an efficient tool for molecular dynamics calculations. *The International Journal of High Performance Computing Applications*, 2019.
18. Stegailov, V., Smirnov, G., Vecher, V.: VASP hits the memory wall: Processors efficiency comparison. *Concurrency and Computation: Practice and Experience* p. e5136. – 2019, <https://doi.org/10.1002/cpe.5136>
19. Polyakov, S., Podryga, V., Puzyrkov, D.: High performance computing in multiscale problems of gas dynamics. *Lobachevskii Journal of Mathematics* 39(9). – P. 1239–1250. – 2018.
20. Ostroumova, G., Orekhov, N., Stegailov, V.: Reactive molecular-dynamics study of onion-like carbon nanoparticle formation. *Diamond and Related Materials* 94, P. 14–20. – 2019.
21. M. Tolstykh, G. Goyman, R. Fadeev, and V. Shashkin. Structure and algorithms of slav atmosphere model parallel program complex. *Lobachevskii Journal of Mathematics*, 39(4):587–595, 2018.
22. V. Akimov, D. Silaev, A. Aksenov, S. Zhluktoy, D. Savitskiy, and A. Simonov. Flowvision scalability on supercomputers with angara interconnect. *Lobachevskii Journal of Mathematics*, 39(9):1159–1169, 2018.

# Вызовы цифровой экономики для науки и образования\*

Л.Н. Щур<sup>1,2</sup>, С.К. Шикота<sup>1,3</sup>

<sup>1</sup> Национальный исследовательский университет Высшая школа экономики

<sup>2</sup> ФИЦ Информатика и управление РАН

<sup>3</sup> Научный центр РАН в Черноголовке

Интенсивная эволюция информационно-коммуникационных технологий в течение последних 30 лет демонстрирует систематический тренд экономических и финансовых акцентов от промышленно-технического к информационно-коммуникационному развитию общества. Эта тенденция в последние годы сформировала такое понятие, как цифровая экономика. Трудно переоценить необходимость проведения исследований как в области научно-технических разработок, так и в области прогнозирования темпов изменения экономики в этом направлении. В настоящей статье мы обсуждаем как уже наметившиеся тенденции в научных исследованиях, направленные на решение возникающих задач, так и вопрос подготовки квалифицированных специалистов, способных успешно работать в различных сферах цифровой экономики.

*Ключевые слова:* цифровая экономика, высшая школа, магистратура, образование

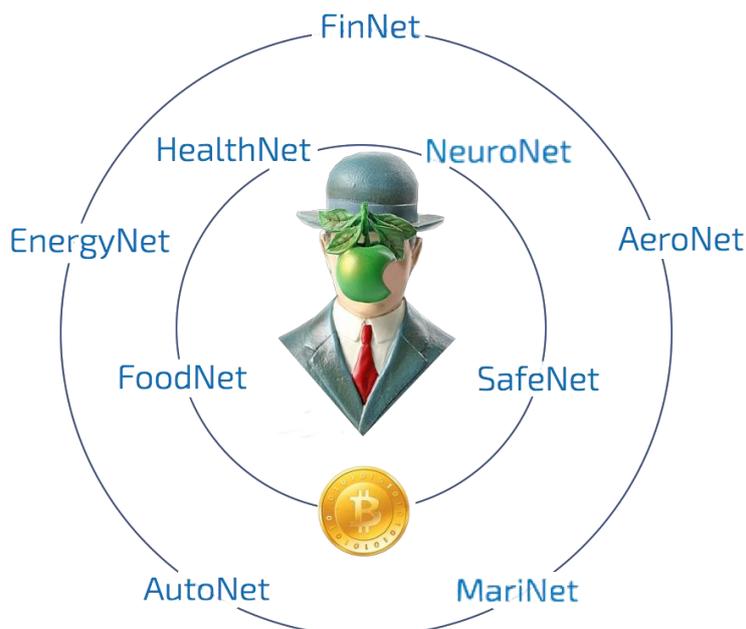
## 1. Цифровая экономика

Термин *цифровая экономика* прочно вошел в ежедневный информационный поток. Трактовка этого нового словосочетания достаточно различна. Объясняется это тем, что при взгляде на возможную деятельность в цифровой экономике, мы увидим различные интерпретации цифровой экономики в зависимости от того, под каким углом к нам повернута плоскость призм восприятия. Таким образом, наличие тех или иных черт в общем спектре этого сложного понятия определяет разнообразие точек зрения на цифровую экономику. В свою очередь, эти точки зрения зависят как от области реализации цифровой экономики, так и от методов информационно-телекоммуникационных технологий (ИКТ), которые для этого применяются.

Области реализации цифровой экономики зафиксированы в Российской Федерации на законодательном уровне в рамках *Национальной технологической инициативы* (НТИ) разделением на девять основных рынков — AeroNet, NeuroNet, EnergyNet, AutoNet, HealthNet, SafeNet, MariNet, FoodNet, FinNet [1]. Эти рынки формируют две оболочки среды цифровой экономики, как схематически показано нами на Рисунке 1. В центре этой среды находится *центральный объект* цифровой экономики – индивидуум. Внутренний круг, *круг индивидуума*, состоит из четырех рынков, которые будут непосредственно соприкасаться с каждым членом общества. На сегодняшний день этот круг находится на начальном этапе формирования. Более развит внешний круг, *круг цифрового информационного общества*, состоящий из пяти рынков, которые определяют уровень приближения глобальной общественной жизни в целом к цифровой экономике. Мы добавили во внутренний круг *виртуальную валюту*, которая пока не имеет законодательных оснований, но которая с большой долей вероятности и, возможно, в каком-то ином виде, будет реализована для применения в повседневной жизни людей.

---

\* Работа выполнена в рамках проекта РФФИ 20-07-00238.



**Рис 1.** Девять рынков Национальной технологической инициативы формируют двойную оболочку индивидуума в цифровой экономике.

Каждый из рынков требует развития как уже существующих технологий, так и разработку новых технологий, создаваемых для конкретной работы на выбранном рынке. Несомненно, что в ходе перехода общества от индустриально-технической экономики к цифровой экономике будут созданы принципиально новые технологии, предсказать которые сегодня невозможно. Ограниченность предсказаний определяется тем, что предсказания основываются на нашем сегодняшнем понимании задач и на уже существующих в данный момент результатах фундаментальных и прикладных исследований, хотя, с очевидностью, развитие цифровой экономики потребует новых научных открытий и технологических разработок, на них основанных. Тем не менее, уже сегодня мы можем смело выделить начатки цифровой экономики в нашем обществе, которые составляют базу для ее развития, и которые основаны на созданных за последние 30 лет информационных и коммуникационных технологиях.

К таким начаткам рынка можно отнести рынок навигации, который составляет базу для развития таких рынков НТИ, как AutoNet, AeroNet, MariNet и FinNet. Этот рынок навигации хорошо знаком каждому владельцу современного мобильного телефона, планшета или ноутбука и предоставляет индивидуумам и предприятиям возможность оптимизации передвижений и идентификации объектов. Его наиболее яркими представителями являются всемирно популярные продукты Яндекс Карты и Google Maps. Эти продукты кардинально изменили рынок перевозок и навигации. Они также изменили качество жизни индивидуума. Этот рынок основан на достаточно широком наборе технологических решений — это определение координат с помощью ГЛОНАСС или GPS, это мобильная связь, это мобильный Интернет, это дружественные интерфейсы пользователя и многое другое, скрытое для глаза пользователей, но над чем работают многие сотни тысяч разработчиков ИКТ технологий и программного обеспечения. Следующее, уже хорошо проработанное и подготовленное к реализации решение ожидает нас на рынке AutoNet. Речь идет о технологии автоматического управления транспортными средствами. Эти технологии включают в себя как упомянутые выше технологии, используемые для навигации, так и технологии компьютерного зрения, искусственного интеллекта и многие другие. Технология опробована в легковых автомобилях Tesla Model S и Tesla Model X. За последние три года произведено более 200 тысяч таких автомобилей [2]. Оборот компании за 5 лет вырос почти в 20 раз. Фирма Tesla уже начала выпуск электро-тягачей Tesla Semi. Поставки планируется начать в 2020 году. Заметим, что внедрение таких тягачей в рынок перевозок изменит как рынок труда автомобильных перевозок, так и экономику рынка AutoNet в целом. Здесь вновь следует обратиться в нашему видению оболочек вокруг индивидуума, которого это внедрение заденет в двух отношениях – массовая потеря рабочих мест водителями (заметим,

что в последние годы Tesla выпускает около 25 тысяч легковых автомобилей ежегодно и способна выпускать примерно столько же электро-тягачей в год) и полный переход на цифровую и роботизированную логистику рынка AutoNet. Иными словами, это внедрение изменит как глобальный рынок AutoNet, так и коснется драматически судеб большого числа водителей и их семей, живущих на обеспечении водителей. Эта социальная проблема уже широко обсуждается в Калифорнии, где наиболее массово используются автомобили Tesla.

В конце 2017 года также произошло знаковое событие, которое возможно условно отнести к рынкам FoodNet и FinNet. Фирма Amazon, крупнейший он-лайн ритейлер, вышла на реальный рынок, создав сеть мини-супермаркетов под брендом Amazon Go [3]. Эти магазины не имеют привычных кассиров за кассой, как нет и кассы, таким образом, нет и привычной для супермаркетов очереди. Вход в магазин обеспечивается за счет приложения для смартфона, который надо приложить к турникету, как при входе в метро. Покупатели кладут выбранные товары в свои сумки, с которыми и выходят. Покупки фиксируются видеокамерами, информацию с которых обрабатывает самообучающееся программное обеспечение, и при выходе через турникет инициируется перевод средств со счета покупателя на счет Amazon. Сравнительно идеальный метод для покупок и продажи, как для покупателя, так и для продавца. Остап Бендер с удовольствием дополнил бы свой список из четырехсот «сравнительно честных способов отъема» денег у населения [4] этим высокоскоростным способом<sup>1</sup>.

Заметим, что, к счастью, процесс входа в цифровую экономику происходит мягко, без революций, и основан на эволюционном процессе развития глобального общества. Даже в этом случае, безусловно, переход будет приводить к определенным сложностям и трудностям, а также породит новые вызовы, но мы не обсуждаем в настоящей статье юридические, социальные и политические аспекты глобального общества, основанного на цифровой экономике.

## 1.1 Взгляд индивидуума на цифровую экономику

Как мы ранее отмечали в нашей статье [5], посвященной роли информационно-коммуникационных технологий (ИКТ) в *глобализации* нашего общества (земной цивилизации), основной вклад ИКТ привел к тому, что исчезло понятие «далекий» в повседневной жизни людей. Если в начале 90-х годов, накануне широкого проникновения в повседневную жизнь новых информационно-телекоммуникационных технологий, круг общения человека был ограничен семьей, работой и общением с друзьями (что в любом случае было ограничено, в основном, границами населенного пункта), то уже сейчас удельный вес общения молодых людей более чем на  $\frac{3}{4}$  лежит вне этого круга. Такое изменение ареала индивидуумов не только стерло понятие дальний, но и перевело значительную часть повседневного общения для большей части активного населения в *виртуальный мир*.

Сфера активности человека обеспечивается программно-техническими ресурсами, в которых все больший вес имеют гостевые книги (Facebook и т.п.) и средства мгновенного общения (Telegram и т.п.), а также более специализированные ресурсы (ResearchGate и т.п.). Возникновение таких ресурсов обусловлено необходимостью поддержки новых видов финансово-экономической деятельности. В свою очередь, такие ресурсы и сами являются важным новым элементом финансово-экономической деятельности. Масштаб этой финансово-экономической деятельности можно понять из того факта, что на весну 2020 более половины мест в первой десятке Forbes занимают владельцы и основатели ИКТ компаний Amazon (1-е), Microsoft (2-е и 8-е), Facebook (4-е), Oracle (6-е) и Google (9-е и 10-е), которые непосредственно связаны с развитием ресурсов, сервисов и инфраструктуры формирующегося сектора цифровой экономики<sup>2</sup>.

Таким образом, цифровое глобальное общество уже народилось в виртуальном мире и активные индивидуумы пользуются им, как данностью

<sup>1</sup> Мы оставляем читателю возможность оправдать использование нами части фразы незабвенного великого комбинатора «сравнительно честных» в применении к владельцам современных технологий.

<sup>2</sup> Весной 2020 ИТ-владельцев была половина – выше приведены данные на 8 июня 2020. За месяцы пандемии произошло значительное финансовое усиление ИТ-сектора.

## 1.2 Взгляд ИКТ-разработчика на цифровую экономику

Разработчик ИКТ-технологий обращает внимание в цифровой экономике на ИКТ технологии, которые и обеспечивают собственно успех развития. Еще 30 лет назад лишь редко кто из специалистов понимал как будет происходить развитие будущих программных систем. Неожиданно для большинства, развитие пошло в направлении применения клонов операционной системы Unix и базируется на идеологии свободно распространяемого программного обеспечения (СПО). В нашей стране основным идеологом этого направления среди профессиональных программистов был академик В.П. Иванников, и мы рекомендуем заинтересованному читателю его лекцию «Что такое СПО» [6]. Мы с коллегами поддерживали это направление, развивая крупнейший в России в 90е и 2000-е годы сервер СПО (смотри, например, [7]). Теперь подход на основе СПО является общепринятым, и такие крупнейшие в мире разработчики программных продуктов, как IBM, Apple и Google, применяют его с большим успехом. Преимущество такого подхода для отечественной экономики мы видим в нескольких аспектах. Во-первых, создание жизнеспособных программных систем сегодня требует сотен и даже тысяч человек-лет работы программистов. Создание же новых систем на основе СПО уменьшает такие затраты в сотни и тысячи раз. Во-вторых, применение СПО дает разработчикам возможность активного участия в его развитии – через определенное время это даст им возможность поддерживать живучесть систем даже в случае увода таких систем из сектора СПО по тем или иным причинам. В-третьих, это дает возможность работать с новейшими технологиями с минимумом затрат на обучение и освоение систем. Яркий пример последнего времени, это система TensorFlow [8], разработанная фирмой Google, которая дает пользователям возможность работы с современными технологиями машинного обучения, при практически минимальных затратах на обучение. В развитии этой программной системы участвуют сегодня сотни тысяч программистов в разных концах света. Заметим, что многие крупные провайдеры информационных услуг уже используют эту систему в своих бизнес-проектах.

Мы не будем перечислять другие программные продукты и ИКТ сервисы, построенные на СПО подходе, которыми пользуются многие сотни миллионов людей в мире.

Цифровая экономика с точки зрения ИКТ разработчика – это набор инструментов, методов и средств использования информационных, вычислительных и коммуникационных технологий во всех сферах: социальной, экономической, промышленной, образовательной и в сфере управления государством на всех уровнях. Они должны предоставлять предсказательный инструментарий, основанный как на широкомасштабном моделировании (как на микро, так и на макро-размерах, обеспечивая, например, разработку материалов с новыми свойствами, в частности, лекарств), так и на извлечении нового знания из анализа данных большого объема. При этом такие новые методы и подходы (блокчейн, рой насекомых и т.п.) следует развивать, анализировать и создавать новые.

## 1.3 Разнообразие точек зрения и взглядов

Мы коротко перечислим другие существующие взгляды на цифровую экономику, которые обсуждаются в нашем обществе. Это взгляд с точки зрения банковского бизнеса, это взгляд с точки зрения работников министерства информационных технологий и связи или министерства науки и образования, это взгляд промышленников, это взгляд разработчиков микроэлектроники, это взгляд политиков, рекламного бизнеса, бизнеса перевозок, и т.п.

Следует вновь обернуться на обсуждаемые события в обществе на рубеже 19-20 веков, в период развития промышленной экономики, чтобы увидеть, как эволюционировали мнения специалистов различных сфер общества. И только по прошествии столетия, на заре новой цифровой экономики, стало возможным провести четкую классификацию и утвердить терминологию промышленного общества. Нам всем предстоит интересный путь к новому знанию. К новому языку терминов и понятий, к новой практике общения и поведения в обществе, как реального, так и виртуального – онлайн-общества.

## 1.4 Фундаментальные научные исследования – их роль в развитии общества

Известно, что научное сообщество наиболее восприимчиво к новшествам. То, что используется сейчас повсеместно, было вовлечено в процесс научных исследований достаточно давно. Например, научные сотрудники не только использовали сеть Интернет с самого начала ее создания в 1991 году, но именно они ее и создавали. Также, как и web пространство сети Интернет, внедренное в 1993 году в ЦЕРНе. Одно из первых *практических* использований новых возможностей реализовали *теоретики*, которые приспособили видео-камеру, чтобы она следила за уровнем кофе в кофеварке.

Только на рубеже 20 и 21 веков Интернет и web технологии начали входить в каждый дом. Иными словами, научное сообщество - это фактически полигон испытаний новых технологий. Именно поэтому следует отдельно обсудить то, что намечается в сообществе, которое занимается фундаментальными научными исследованиями.

Ожидаемый прорыв в развитии фундаментальных исследований связывается с реализацией методологии «Большие Данные» с использованием подхода «Наука с интенсивным использованием данных» (Data Intensive Scientific Discovery). Мы находимся на пороге возникновения новой парадигмы научных исследований в дополнение к уже устоявшимся подходам получения нового знания путем эксперимента, теории и компьютерного моделирования.

Наиболее яркий и общеизвестный пример - это обработка данных с Большого адронного коллайдера (БАК) в ЦЕРНе, которая привела к экспериментальному обнаружению бозона Хиггса, элементарной частицы с энергией 125 гигаэлектронвольт. Организация обработки данных – это самый крупный научный ИКТ проект. Затраты на создание инфраструктуры по передаче, хранению и обработке данных составили более половины затрат на поиски бозона Хиггса. Технология Грид развернута поверх научных компьютерных сетей стран-участниц и использует специальные каналы связи между странами. Это первый пример из разряда Big Data. И этот пример привел к несомненному успеху.

Второй выдающийся пример реализован фирмой ИБМ. В научно-исследовательской лаборатории Accelerated Discovery Lab (ADLab) фирмы ИБМ в Алмадене (Калифорния, США) развернут программно-аппаратный комплекс «ИБМ Ватсон» для хранения и обработки больших данных, а также для проведения фундаментальных исследований в области интенсивной обработки данных. «ИБМ Ватсону» принадлежит серьезное достижение в области поиска веществ, ответственных за рост злокачественных клеток – за полгода исследований с использованием методологии анализа больших данных, хранящихся в центре Алмадена, были обнаружены два фермента, влияющих на протеин p53, ответственный за процесс активации роста злокачественных клеток. Для сравнения, обнаружение новых ферментов с такими свойствами делается всеми лабораториями мира с частотой один фермент в год. Именно этот пример является переломным в оценке перспектив научных исследований в области извлечения нового знания из данных большого объема.

Третий пример – это интенсивные исследования по созданию программных и аппаратных средств, которые проводятся фирмой Google совместно с фирмой Nvidia. В рамках проекта Google Brain [9] создается программное обеспечение по работе с данными, а фирма Nvidia запустила новую серию графических ускорителей Volta под эту программу. Этот пример пока без видимых достижений, которые, тем не менее, есть основания ожидать уже в недалеком будущем. Причина оптимизма в том, что проект основан на концепции СПО, и в его развитие включена большая армия исследователей и инженеров по всему миру.

Извлечение нового знания из больших данных находит применение не только в области фундаментальной науки, где проходят апробацию новые методы и алгоритмы обработки данных, но и во всех областях деятельности, включая финансовые рынки, коммерцию, социальные исследования, а также в прогностических исследованиях.

Новое знание, полученное в ходе изучения больших данных, может представлять собой математические модели функционирования сложных систем: кибернетических, технических и социальных; получения прогностических знаний и моделей, пролегоменов новых прикладных наук и технологий, касающихся не только информатики, но и естественных, инженерных, экономических и социальных наук.

Высока актуальность фундаментальных исследований и для систем мониторинга и управления системами добывающего, обрабатывающего и топливно-энергетического комплекса, а также коммунальных систем с точки зрения принципиального повышения качественных и количественных показателей их работы.

Выполнение ориентированных фундаментальных исследований в этом направлении возможно путем консолидации опыта, знаний, наработок и общего научного потенциала коллективов и специалистов, активно работающих в высших учебных заведениях и научно-исследовательских организациях России.

Безусловно, что именно технологии, разрабатываемые научным сообществом и будут определять темп развития цифровой экономики.

## 2. Подготовка кадров - задача высшей школы

Для успешной реакции общества РФ на новые вызовы необходимы квалифицированные кадры во всех сферах цифровой экономики. Здесь уместно поделиться идеями о нашем подходе к решению такой задачи в МИЭМ НИУ ВШЭ (Московский институт электроники и математики, Национальный исследовательский университет «Высшая школа экономики»).

Мы нередко слышали тезис, что для успешного решения задач цифровой экономики нужны специалисты по внедрению и использованию новых технологий. Специалисты, которые хорошо знают, какую нужную нажать кнопку или выбрать правильную клеточку для клика мышкой. Которые знают, какой и у кого заказать программный продукт. Безусловно, такие специалисты нужны, и их надо много. При этом, однако, выпадает из поля зрения то обстоятельство, что те самые нужные программные продукты откуда-то берутся. Кроме того, следует учесть, что доступ к новым программным продуктам может быть и ограничен в силу разных обстоятельств. Иными словами, нам также нужны специалисты, которые способны эти новые программные продукты создавать.

Новая магистерская программа «Суперкомпьютерное моделирование в науке и инженерии» [10], обучение по которой началось с 1 сентября 2018, имеет своей целью подготовку вышеупомянутых специалистов. Как разработчиков, так и квалифицированных пользователей. Каковы же основные идеи?

Первая идея – это применение технологии подготовки высококвалифицированных исследователей, родоначальником которой был Абрам Федорович Иоффе. Это технология активного обучения, с ранним привлечением студентов к научной работе, с совместным изучением новых научных статей на семинарах и с тесным научным руководством. К школе академика Иоффе можно отнести нобелевских лауреатов П.Л. Капицу, Н.Н. Семенова, И.Е. Тамма и Л.Д. Ландау, трижды героев Социалистического труда, создателей ракетно-ядерного щита нашей страны И.В. Курчатова, А.П. Александрова и Я.Б. Зельдовича и многих других физиков с мировым именем. Школа академика Иоффе – это известная всему миру плеяда физиков-экспериментаторов и физиков-теоретиков. Сильнейшая в мире научная школа по физике. Абрам Федорович основал физико-технический институт, который сегодня носит его имя. Всему миру известен сотрудник этого института академик Алферов, нобелевский лауреат, создатель технологии гетероструктур, лежащей в основе современной микроэлектроники. Абрам Федорович создал также физико-математический факультет Петербургского политехнического института для подготовки инженеров-физиков.

Технология обучения была развита его учениками, которые основали МФТИ, Московский физико-технический институт, ныне – Исследовательский университет, ведущий ВУЗ страны, да и всего мира, по подготовке физиков для науки.

Задача, которую мы ставим в нашей магистерской программе, это подготовка специалистов по разработке методов и подходов по созданию программных систем, применимых на суперкомпьютерах. Что такое суперкомпьютеры и зачем они нужны? Уровень суперкомпьютеров определяет степень технологического развития государства. Исторически их создание приходится на конец 40-х – начало 50-х годов, и они были частью программы по разработке ядерного оружия и ракетной техники. Следует заметить, что сегодня разработка ядерного оружия ведет-

ся в развитых странах исключительно с использованием суперкомпьютерной техники – натурные испытания такого оружия запрещены уже более полувека.

Для чего еще используются суперкомпьютеры? Для моделирования погоды, для разработки новых лекарственных веществ, для создания новых материалов. В рамках нашей программы мы делаем акцент на научные исследования. Почему? Если Вы прочитали предыдущий раздел, то заметили основной тезис – новое испытывается в научном сообществе. Конечно, для дальнейшего обучения в аспирантуру пойдет 20 процентов магистров. Остальные 80 процентов с полученным багажом знаний, умений, навыков и изученных методов, подходов и алгоритмов без труда найдут себе работу на широком рынке труда – в производстве и проектировании, в банках и финансах, в управлении и ИТ-компаниях. Такая уверенность основана на имеющемся у нас опыте подготовки специалистов в упомянутом выше физтехе.

Основа обучения состоит в комбинации прикладной математики, программировании и методах обработки больших данных со статистической физикой и квантовой механикой. Студенты осваивают основные подходы математической физики и модели статистической механики. Это дает им возможность широкого взгляда на проводимые ими исследования в различных областях знаний. Возможная тематика выполняемых дипломных работ широка. Это может быть анализ методов квантовых вычислений, или применение фундаментальных теорем математики для разработки алгоритмов машинного обучения, или применение методов машинного обучения для оптимизации работы молочных ферм, или анализ статистических данных по проблеме алкоголизма или проблеме финансового неравенства. При этом, студенты получают возможность использовать полученный багаж в указанных областях, что неизбежно должно привести к успеху.

Такая уверенность основана на точном замечании выдающегося физика-теоретика Александра Полякова (прошедшего школу на физтехе, аспирантуру в Институте теоретической физики им. Л.Д. Ландау, ныне работающего в университете в Принстоне), которое он сделал в интервью после вручения ему очередной высокой премии. Оно состоит в том, что совершенно различные явления в природе описываются одними и теми же уравнениями, потому что явлений в природе много, а хороших уравнений – мало. Поэтому, знание относительно небольшого числа уравнений, методов анализа и алгоритмов, которому мы учим наших студентов, помогут им решать практически любые задачи в практически любой области науки, экономики, общественной жизни и финансах.

Акцент на суперкомпьютеры – это акцент на технологию вычислений будущего. Дело в том, что производительность одного вычислительного элемента, изготовленного по современной технологии, достигла предела. Для увеличения производительности обработки информации развитие идет в направлении увеличения числа вычислительных элементов, которые называются ядрами (cores). Современные суперкомпьютеры имеют миллионы ядер, но далеко не все разработанные программы способны одновременно использовать такое количество ядер. Наше обучение построено на изучении новых подходов и методов, которые применяются для разработки алгоритмов, и построении программных систем, которые способны работать на большом числе ядер, которые способны эффективно обрабатывать информацию на суперкомпьютере. В том числе, для решения проблем машинного обучения.

Задача подготовки состоит в передаче студентам опыта создания таких подходов, алгоритмов и программных систем. В работе над научными проектами. В приобретении опыта работы на суперкомпьютерах. В приобретении опыта применения полученных знаний в тех областях, которые интересны студентам.

### **3. Заключение**

Мы надеемся на то, что наша статья вызовет продуктивную дискуссию по вопросам развития фундаментальных исследований и подготовки научных кадров, которые с успехом будут развивать исследования и применения новых технологий в интересах развития цифровой экономики страны.

Начало 2020 года ознаменовалось проблемой глобального масштаба, пандемией COVID-2019, которая перевела миллиарды людей в режим самоизоляции. Процесс обучения это не ос-

тановило. Учебные заведения перешли в режим удаленного обучения студентов. Надо заметить, что Высшая школа экономики, которая в последние годы интенсивно развивала методику и технологии удаленного обучения, быстро перевела работу в режим online. Этому помогли также имеющиеся обширные связи с мировыми лидерами как в области удаленного обучения, так и в области технологий.

К началу конференции будет накоплен опыт и статистика использования, которые безусловно будет интересно обсудить и проанализировать.

## Литература

1. Национальная технологическая инициатива, Учебный курс. <http://www.nti2035.ru/> (дата обращения 17.04.2020 г.)
2. <http://ir.tesla.com/> (дата обращения 05.06.2020 г.)
3. Amazon Go, <https://www.amazon.com/b?node=16008589011> (дата обращения 17.04.2020 г.)
4. И. Ильф и Е. Петров, «Золотой теленок», гл. 2, (1931).
5. Л.Н. Щур, Роль инфокоммуникационных технологий в развитии процесса глобализации научных исследований, [Информационное общество, №5, 16-24 \(2012\)](#).
6. В.П. Иванников, Что такое СПО. Механика, управление и информатика, №5, 105-109 (2011) <https://elibrary.ru/contents.asp?issueid=1006665>.
7. Крашаков С.А., Щур Л.Н. FTP.Chg.RU - 10-летняя история крупнейшего в России архива свободнораспространяемого программного обеспечения и дальнейшие перспективы, Тр. Всероссийской научной конференции "Научный сервис в сети Интернет ' 04" (Новоросийск, 20-25 сентября 2004 г.), стр. 250-254.
8. TensorFlow, <https://www.tensorflow.org/> (дата обращения 17.04.2020 г.)
9. Google Brain, <https://research.google.com/teams/brain/> (дата обращения 17.04.2020 г.)
10. <https://www.hse.ru/ma/supmod/> - Магистерская программа «Суперкомпьютерное моделирование в науке и инженерии» МИЭМ НИУ ВШЭ (дата обращения 18.04.2020 г.).

# Генерация тетраэдральных сеток для суперкомпьютерного моделирования обтекания аэрокосмических объектов\*

М.К. Ермаков

Институт проблем механики им. А.Ю. Ишлинского РАН

Для генерации сверхподробных тетраэдральных сеток объемом до 1 миллиарда ячеек используется открытое ПО Gmsh. Пакет позволяет строить автоматически указанные сетки на ПК в параллельном режиме OpenMP за время не превышающее 1 часа при использовании ПК Intel i7-9700K. Описан опыт применения пакета для построения пространственных сеток для задач обтекания аэрокосмических объектов. Приведены примеры суперкомпьютерных расчетов обтекания аэрокосмических объектов.

*Ключевые слова:* генерация сеток, тетраэдральные сетки, Gmsh, open source, суперкомпьютерное моделирование, обтекание аэрокосмических объектов

## 1. Введение

Построение пространственных сеток в автоматическом режиме и сеток сверхвысокого разрешения для суперкомпьютерного моделирования, в частности, обтекания аэрокосмических летательных аппаратов (ЛА) и объектов является актуальной научно-технической задачей. При кажущейся простоте постановки задача генерации сетки является достаточно сложной и ресурсоемкой даже при использовании простейших пространственных элементов в виде тетраэдров. Используемые до недавнего времени программные средства были существенно ограничены автоматической генерацией ячеек сетки, достигающих десятков миллионов ячеек, и быстродействию.

Программные продукты по генерации сеток, как правило, функционируют в связке с системами автоматизированного проектирования, и являются частью программных сред ANSYS, Fluent, CFX, Numeca и других, а также в виде отдельных пакетов ICEM, PointWise, Simmetrix и других. Открытые пакеты представлены Gmsh, EnGrid, Netgen, Tetgen, Ani3D [1] и входящими в OpenFOAM программами BlockMesh, PolyDualMesh и snappyHexMesh.

Сложность и состояние программных систем для построения сеток в настоящее время таковы, что, условно говоря, они похожи не на автомобили, для которых пересечь с одного на другой не составляет труда, а на самолеты, для управления каждым из которых требуется отдельное специализированное обучение. В силу данного фактора использование и сравнение нескольких пакетов является проблематичным. В связи с направлением импортозамещения коммерческих зарубежных продуктов были исследованы открытые программные продукты (open source) построения тетраэдральных пространственных сеток, допускающих генерацию в полностью автоматическом режиме. В работе представлен опыт использования пакета Gmsh, наиболее функционального и производительного среди других.

## 2. Общее описание и характеристики Gmsh

Программный комплекс Gmsh (версия 4.5.6) [2-4] предназначен для автоматической генерации неструктурированных поверхностных и пространственных сеток для геометрических объектов, заданных своими поверхностями. Набор поверхностей задается как на основе собственного упрощенного геометрического ядра, так и на основе геометрического ядра и программных средств Open Cascade. Система Gmsh представляет собой интерактивную среду, включающую геометрическую модуль в виде сильно упрощенной подсистемы САПР, модуль генерации сеток и модуль визуализации конечно-элементных данных. Использование развитой открытой системы Open Cascade позволяет использовать широкий круг функций геометриче-

---

\* Работа выполнялась при финансовой поддержке гранта РФФИ 19-01-00515

ского ядра: полный набор примитивов поверхностей и объемов, набор булевских операций над геометрическими объектами, параметризацию и документирование процесса построения объекта и создаваемой сетки в виде скрипта, систем импорта геометрии в стандартных форматах и экспорта геометрии.

Помимо поверхностных треугольных сеток пакет поддерживает преобразование их в частично четырехугольные элементы, а также образование на их основе призматических элементов при преобразовании выталкивания геометрии и сетки. Сетки, однако, в любом случае остаются неструктурированными.

Набор алгоритмов построения сеток базируется на сочетании метода Делоне, метода продвигаемого фронта и методов разбиения и перестроения сетки. Методы характеризуются различной степенью качества построения сетки, быстродействием и степенью надежности (робастности, безотказности), обозначающей возможности создания сетки для геометрии сложного типа.

Пакет написан полностью на алгоритмическом языке C++ и функционирует в интерактивном режиме и в пакетном режиме (командной строки) в операционных системах Windows/Linux/macOS. Посредством использования API с подключением библиотек пакет доступен в алгоритмических языках C/C++, Fortran, Python и Julia. Открытый код содержит порядка 500 тыс. строк исходного текста, система имеет примерно 8000 загрузок в месяц со своего сайта и 500 цитирований в год в литературе. Разработанная учеными и программистами для собственного использования система удачно сочетает функциональность и удобство работы. Пакет Gmsh развивался в течение 20 лет и продолжает активно развиваться авторами, имеет подробную документацию с примерами и поддержку разработчиков в виде форума.

### 3. Методы построения сеток в Gmsh

Используется несколько подходов к заданию локального размера создаваемой сетки. В первом подходе создается сетка, близкая к равномерной, с указанием минимального и максимального размеров сетки. Во втором подходе каждой точке (вершине) топологии геометрического объекта приписывается дополнительный параметр, определяющий локальный линейный размер сетки. На топологических элементах объекта локальный линейный размер элементов интерполируется линейной функцией. Помимо задания линейного размера сетки возможно ограничение величины шага сетки по угловому размеру элемента, определяемого кривизной поверхности. Третий подход заключается в задании дополнительных областей простой формы (шар, параллелепипед), в которых задается локальный линейный размер сетки.

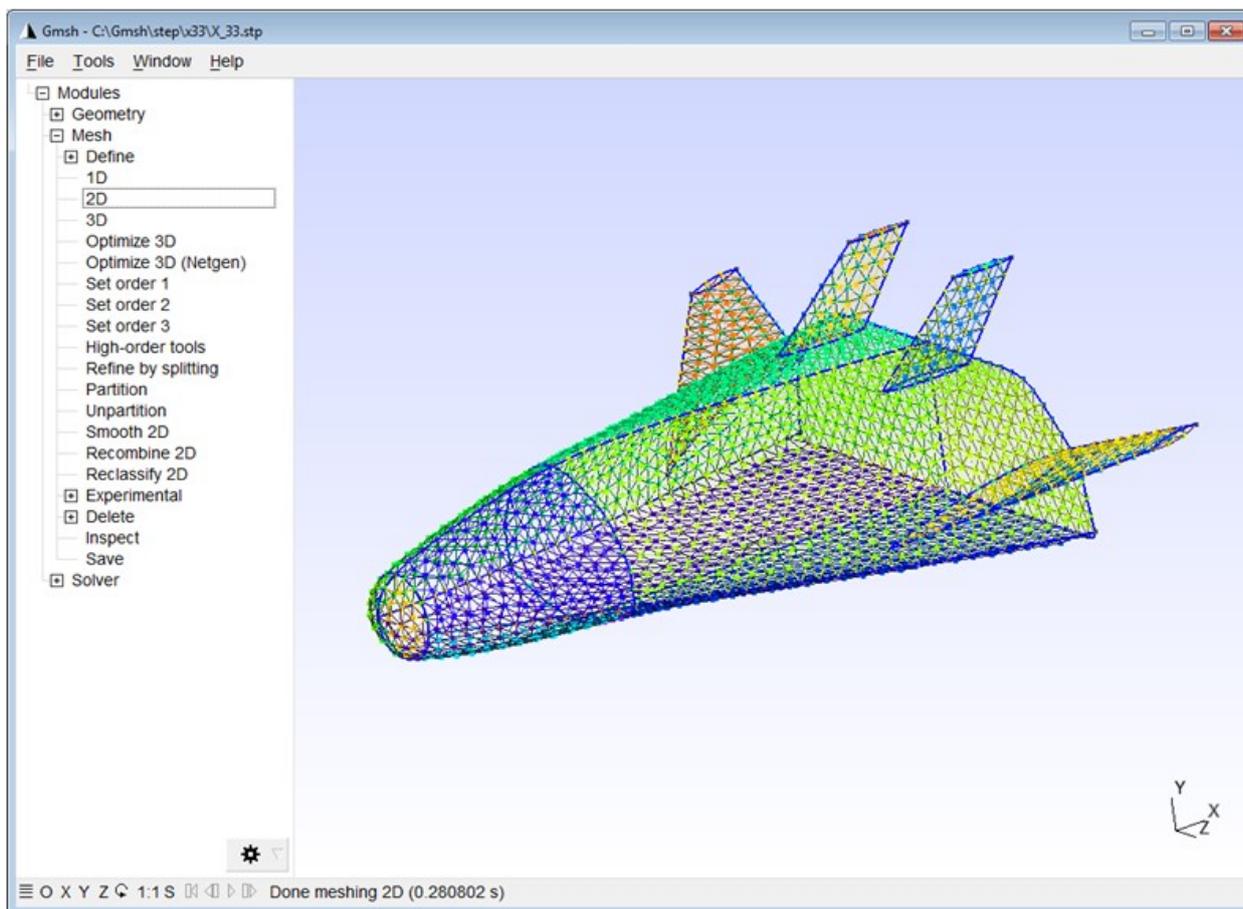
После задания локального шага система последовательно, по командам пользователя, строит одномерную сетку вдоль ребер объекта (1D), двумерную сетку на поверхностях объекта (2D) и трехмерную сетку внутри объемов объекта (3D). По желанию пользователя каждая сетка в виде элементов, таких как, вершины, ребра, грани и тетраэдры с их номерами или без номеров могут визуализироваться. Также возможна визуализация отдельных элементов или их групп, относящихся к любому виду элементов топологии. После этапа генерации двумерной сетки возможно улучшение качества сетки, число итераций сглаживания задается. После этапа построения трехмерной сетки возможна оптимизация сетки с целью улучшения качества элементов.

В качестве алгоритмов построения тетраэдральной пространственной сетки используются 3-D варианты метода Делоне для алгоритмов Delaunay [5, 6] и НХТ [3], метод продвигаемого фронта J. Schoeberl для алгоритма Frontal [7]. Собственная оптимизация Gmsh тетраэдральных сеток построена на перестановках ребер и граней с перемещением вершин [2]. Дополнительно подключаемая оптимизация Netgen основана на сочетании локального метрического сглаживания и топологических преобразований [7].

## 4. Результаты

### 4.1 Работа в системе в интерактивном режиме

Перейдем к практическим результатам построения сеток и описанию полученного опыта работы с системой Gmsh. Внешний вид системы в интерактивном режиме с меню древовидной структуры, геометрией объекта, представленной ребрами геометрии, и двумерной сетки ЛА X-33 представлен на рис. 1. Меню представлено действиями из раскрытого раздела Mesh. Благодаря классическому горизонтальному меню возможен выбор параметров сгущения сетки, алгоритмов построения поверхностной и пространственной сеток, параметрам визуализации элементов сетки, данным по использованию ресурсов для построения сетки и другим параметрам. Позиции (действия) древообразной структуры являются набором команд пользователя для выполнения функций системы. В окне системы представлены ребра геометрии объекта и поверхностная сетка, которая по умолчанию рисуется разным цветом на каждой поверхности. В целях наглядности сетка выбрана с малым количеством треугольников на поверхности и без области обтекания, которая строится вокруг ЛА. В строке статуса обозначены выполненная операция – построение двумерной сетки и время выполнения операции. Для контроля действий в процессе выполнения операций возможно открытие окна сообщений, в которое выводится информация о ходе генерации сетки для элементов объекта.



**Рис. 1.** Общий вид интерактивной оболочки Gmsh с деревом меню построения сетки и примером геометрии и поверхностной сетки для X-33

## 4.2 Работа в системе в режиме командной строки

Для использования пакета с минимальным использованием дополнительных ресурсов и в пакетном режиме имеется не интерактивный режим работы в командной строке. Для его использования параметры алгоритмов и выполняемые команды задаются в загружаемом скрипте с расширением `geo` и в параметрах командной строки вызова пакета `gmsh`. В данном режиме построение и оптимизация сеток происходит полностью автономно.

## 4.3 Построение пространственных сеток для обтекания аэрокосмических объектов

В качестве примеров аэрокосмических объектов для демонстрации работы системы используются модели командного отсека (спускаемой камеры) Apollo, ЛА X-33 и космического аппарата Space Shuttle. Геометрии примеров объектов представлены на рис. 2.

Для получения сетки для моделирования обтекания ЛА необходимо создание области течения вокруг объекта. Размеры области и её расположение зависят параметров и характера обтекания, поэтому её желательно создавать отдельно от создания геометрической модели ЛА. Для создания области обтекания создавалась кривая в виде сплайна для границы входа потока и примыкающая прямая для выходной границы течения. Далее кривая в виде этих двух элементов выталкивается по углу на 180 градусов (обычно рассматривается обтекание половины аппарата в силу его зеркальной симметрии), образуя объем вращения вокруг ЛА. Вычитая из полученного объема исходный объем летательного аппарата, получаем необходимую нам область обтекания, в которой и необходимо построить пространственную сетку.

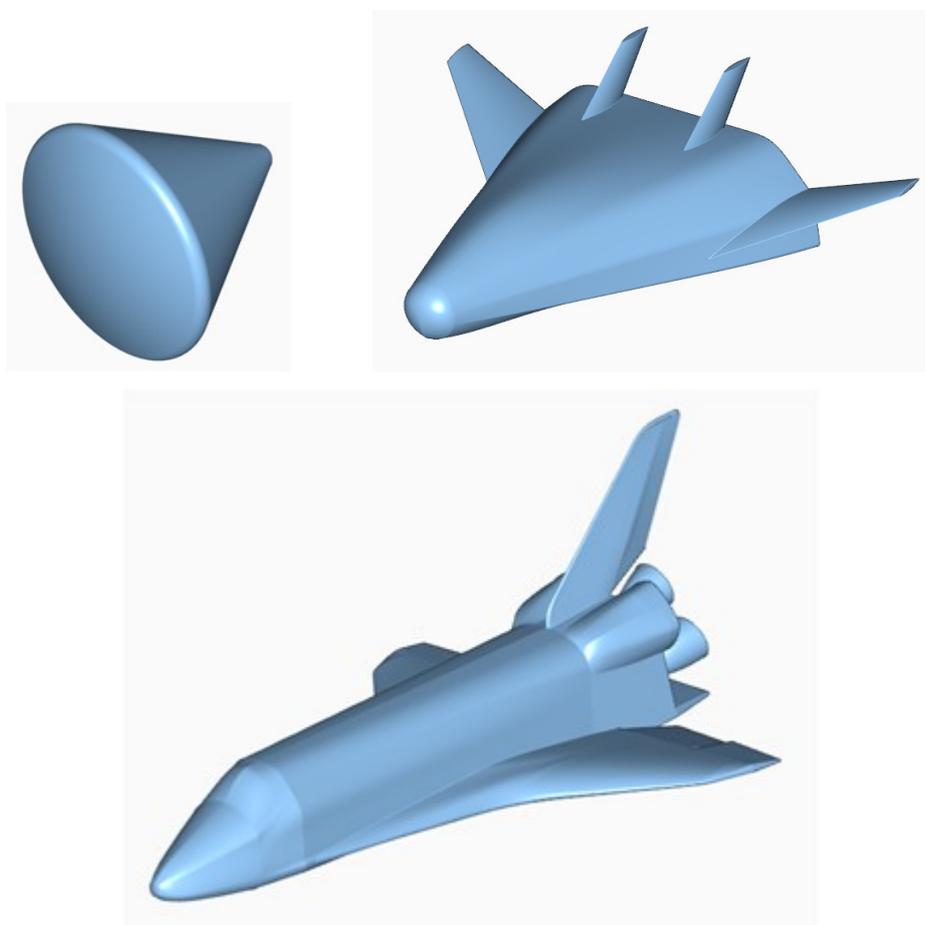


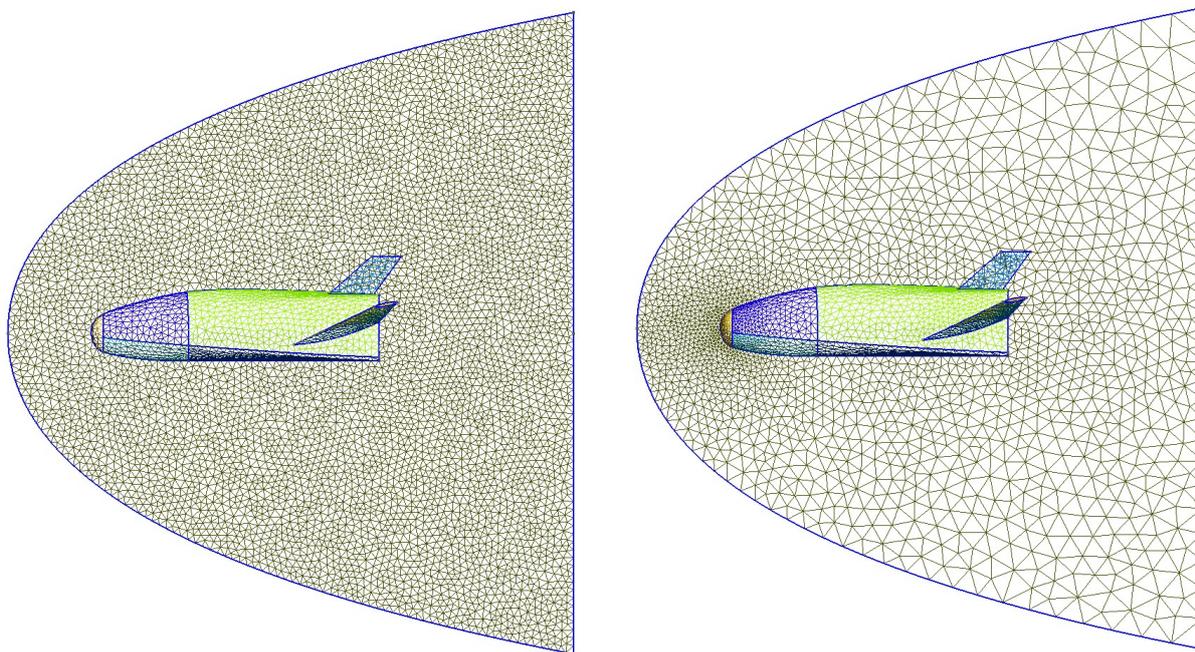
Рис. 2. Геометрия моделей Apollo, X-33 и Space Shuttle

Весь описанный выше набор операций построения области обтекания записывается в виде скрипта длиной примерно 15 строк, который можно будет использовать каждый раз для автоматизации действий, внося в него при необходимости изменения параметров. Примеры области обтекания вместе с построенной сеткой в области симметрии можно видеть на рис. 3.

Для построения сеток целесообразно использовать неравномерные сетки со сгущениями в областях наибольших градиентов решения, управляя распределением локального размера ячеек заданием этого размера в вершинах геометрического объекта. Пример использования равномерной сетки и неравномерной сетки с заданием нескольких характерных линейных размеров сетки представлен на рис. 3. Использование неравномерных сеток позволяет существенно уменьшить количество расчетных ячеек, используя более крупные ячейки там, где наличие мелких ячеек не является обязательным. Опыт построения сеток показал, однако, что различие ячеек сетки с отношением линейных размеров более чем 10 является нецелесообразным. Отметим, что при отношении линейных размеров ячеек в 10 раз, объемы таких ячеек различаются в 1000 раз.

Для исследования возможностей пакета тестировался весь набор имеющихся в его составе алгоритмов с точки зрения качества получаемой сетки, быстродействия и использования оперативной памяти. Для генерации поверхностной сетки используются алгоритмы Mesh-Adapt, Delaunay и Frontal-Delaunay. Для наиболее сложных поверхностей наиболее предпочтительным считается Mesh-Adapt, наиболее быстрым – Delaunay и генерирующим наиболее качественную сетку - Frontal-Delaunay. Опыт использования подтвердил, что Delaunay является наиболее быстрым из них, в то время как другие два алгоритма могли существенно замедляться на отдельных поверхностях.

Для генерации трехмерных сеток используются также три алгоритма: Delaunay, Frontal и НХТ. Алгоритм Delaunay считается наиболее безотказным, а алгоритм НХТ – наиболее быстрым, является новой и эффективной реализацией параллельного алгоритма Delauney.



**Рис. 3.** Примеры равномерной и неравномерной поверхностных сеток в области симметрии и на поверхности ЛА X-33

Оценки производительности и необходимой оперативной памяти наиболее быстрого алгоритма НХТ построения трехмерной сетки представлены в табл. 1 для сеток объемом от 3 млн. ячеек до 1 млрд. ячеек на ПК Intel i7-9700K. Время, необходимое только для генерации трехмерной сетки, для сетки размером 3 млн. ячеек составляет примерно 5 секунд, в то время как для сетки в 170 млн. ячеек время генерации сетки составляет менее 5 минут. Требуемая опера-

тивная память для этих сеток составляет 0.5 ГБ и 24 ГБ, что доступно на стандартных современных высокопроизводительных ПК.

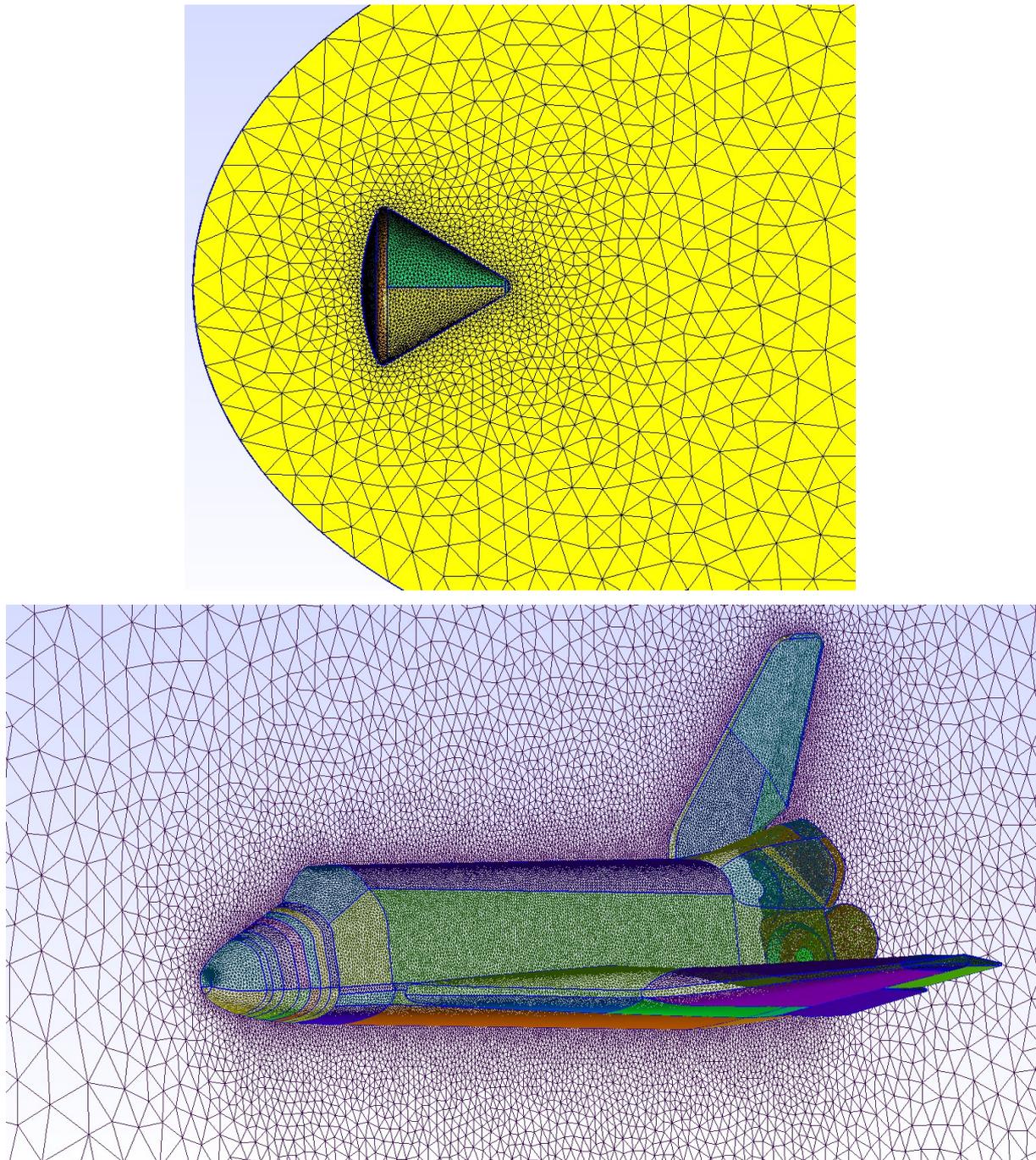
**Таблица 1.** Ресурсы для построения трехмерной сетки различного объема методом НХТ

Размер сетки, ячеек	Оперативная память	Время построения
3 М	0.5 ГБ	5''
10.5 М	1 ГБ	16''
20.8 М	2.4 ГБ	32''
50 М	6.7 ГБ	74''
86 М	11.6 ГБ	128''
170 М	24 ГБ	255''
1000 М	< 128 ГБ	< 3600''

Теперь приведем более полное исследование производительности и требуемой оперативной памяти для остальных методов генерации трехмерной сетки и двух методов оптимизации сетки. В табл. 2 представлены эти данные для сетки объемом в 50 млн. ячеек. Отметим, что наиболее быстрый алгоритм НХТ содержит внутри себя оптимизацию сгенерированной трехмерной сетки. Анализ результатов в табл. 2 показывает, что новый, появившийся в конце 2019 г. алгоритм НХТ существенно, более чем в 10 раз, опережает по скорости генерации сетки другие два метода; также алгоритм НХТ имеет в два раза «меньший аппетит» в точки зрения оперативной памяти. Обычная оптимизация выполняется очень быстро по сравнению с методами Delaunay и Frontal, но требует в полтора больше оперативной памяти. Оптимизация Netgen реализует существенно более качественную оптимизацию сетки, но требует времени, сравнимого с временем генерации сетки.

**Таблица 2.** Ресурсы для построения и оптимизации трехмерной сетки объемом 50 М ячеек

Методы	Оперативная память	Время построения
Delaunay	12.6 ГБ	23'
Frontal	15.3 ГБ	52'
НХТ + Optimization	6.7 ГБ	1.3'
Optimization Gmsh	20.4 ГБ	1.3'
Optimization Netgen	15.4 ГБ	48'



**Рис. 4.** Примеры поверхностных сеток в области симметрии и на поверхностях моделей Apollo и Space Shuttle

С точки зрения производительности и требований к размеру оперативной памяти алгоритм НХТ существенно превосходит два остальных алгоритма.

На рис. 4 дополнительно показаны трехмерные сетки, сгенерированные для образцов аэрокосмических аппаратов Apollo и Space Shuttle, в области симметрии и на поверхностях аппаратов. Модель Space Shuttle содержит существенно большее количество геометрических поверхностей (показаны на рисунке разным цветом), однако система успешно справилась с генерацией сетки.

#### 4.4 Оценка качества генерируемых сеток

Теперь исследуем качество сгенерированных сеток. Встроенной в систему оценки качества сетки найти не удалось, поэтому пришлось написать небольшой код. В качестве критерия качества элемента сетки рассмотрим параметр  $q$ , равный умноженному на 3 отношению радиуса вписанной в тетраэдр сферы ( $r$ ) к радиусу описанной сферы ( $R$ )

$$q = 3 * r / R. \quad (1)$$

Этот параметр не превышает единицы для любых тетраэдров, причем значения единицы он достигает для тетраэдра, имеющего все равные между собой ребра и, соответственно, грани. Радиусы вписанной в тетраэдр и описанной сферы вычисляются по формулам:

$$r = 3 * Vol / (S_1 + S_2 + S_3 + S_4), \quad (2)$$

$$R = Surf / 6 * Vol, \quad (3)$$

где  $Vol$  – это объем тетраэдра ( $1/6$  смешанного произведения векторов, исходящих из одной вершины),  $S_1, S_2, S_3$  и  $S_4$  – площади граней тетраэдра,  $Surf$  – площадь треугольника со сторонами  $(a_{12} \times a_{34}, a_{13} \times a_{24}, a_{14} \times a_{23})$ , где  $a_{ij}$  – есть длина ребра, соединяющая точки  $i$  и  $j$  тетраэдра.

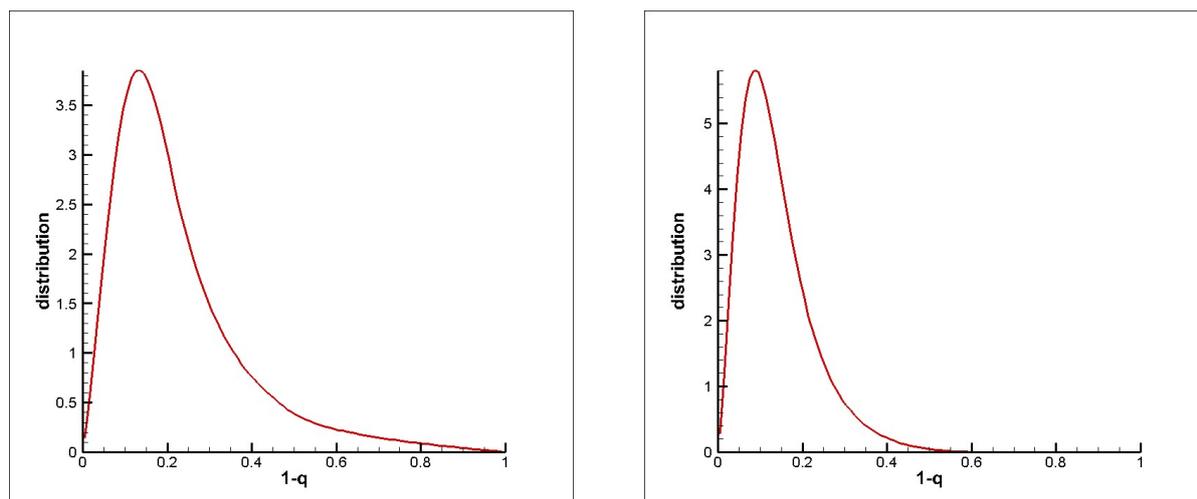


Рис. 5. Распределение функции  $1-q$  для пространственных сеток после генерации сетки и после оптимизации пакетом Netgen

Площадь треугольника со сторонами  $a, b$  и  $c$  может быть вычислена по формуле Герона

$$S = p * (p-a) * (p-b) * (p-c), \quad (4)$$

где  $p = (a+b+c)/2$  является половиной периметра треугольника.

Для оценки качества будем оценивать распределение величины  $1-q$  по удельному весу ячеек. Чем больше вес ячеек с высоким качеством, тем больше кривая распределения прижимается к оси ординат, чем хуже качество сетки, тем больше значения функции распределения при значении 1 на оси абсцисс. На рис. 5 представлены распределения качества тетраэдров после генерации сетки методом Delaunay (слева) и после её оптимизации алгоритмом пакета Netgen.

На графике распределения после генерации сетки (слева) большая часть функции распределения расположена в левой части, соответствующей ячейкам с высоким качеством, однако виден и заметный хвост в правой части графика, при значениях больше 0.5, соответствующей

ячейкам низкого качества. При проведении простой и быстрой оптимизации сетки изменения в функции распределения заключаются в «обрезании» функции распределения при значениях аргумента больше 0.6. При оптимизации Netgen, представленной в виде функции распределения на правом графике, заметно увеличивается доля ячеек в высоком качестве, а доля ячеек с низким качеством существенно стремится к нулю, начиная со значений аргумента 0.4.

Практически важным для моделирования является и минимальный размер ячейки, оцениваемый по радиусу вписанной в тетраэдр сферы. При генерации пространственной сетки объемом 5 млн. ячеек методом Delaunay с минимальным размером ребра ячейки 25 минимальный радиус вписанной сферы составлял 0.035, после оптимизации сетки он увеличился до 0.23, а при оптимизации сетки алгоритмом Netgen достигает значения 2.9. При генерации пространственной сетки методом НХТ, включающим внутреннюю оптимизацию, минимальный радиус вписанной сферы составил 4.2, что лучше, чем в предыдущих случаях.

#### 4.5 Исследование параллельного метода генерации сетки

Явным достоинством новой реализации метода Делоне в алгоритме НХТ состоит в возможности использования параллельного режима OpenMP. Для его использования необходимо перекомпилировать и пересобрать пакет Gmsh с ключом ENABLE\_OPENMP согласно документации пакета. В связи с тем, что родной средой для сборки пакета является Linux-подобная среда, это было сделано в Ubuntu 18.04 LTS с установленным пакетом OpenCascade.

Результаты тестов при использовании от одного до 8 физических ядер процессора i7-9700K представлены в табл. 3.

Таблица 3. Ускорение за счет использования многоядерности (OpenMP)

Количество ядер	Время построения	Ускорение
1	74"	
2	53"	1.4x
4	39"	1.9x
8	10"	2.4x

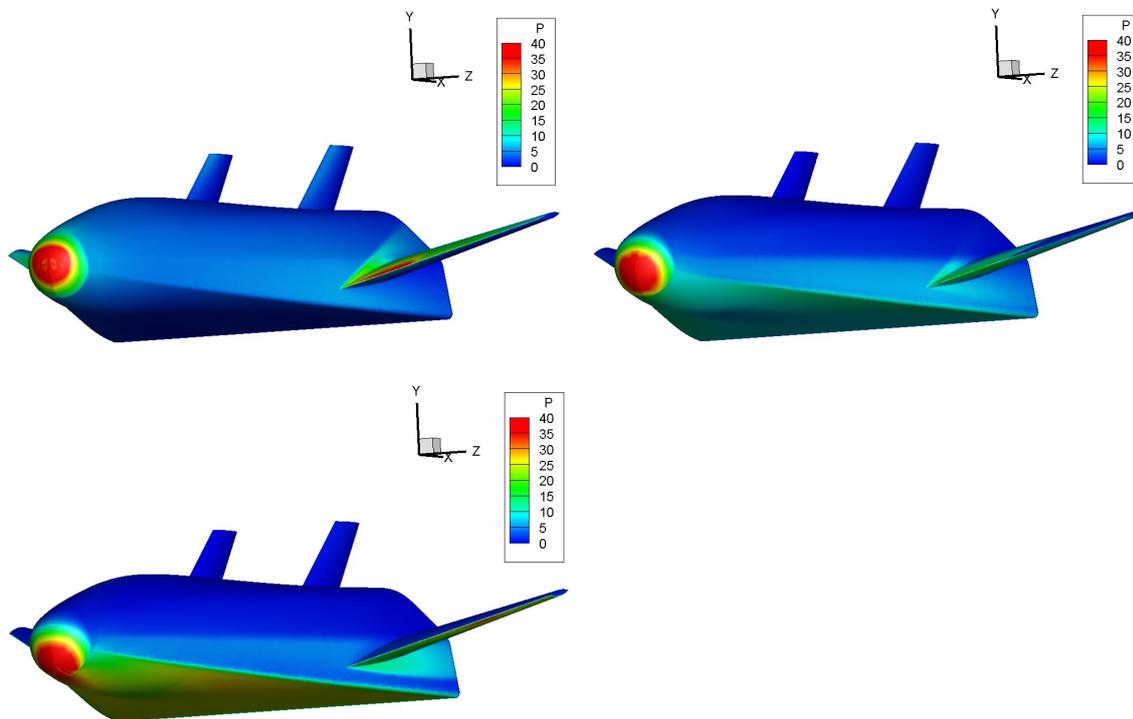
Максимальное ускорение при использовании 8 физических ядер достигало значения 2.4, что объясняется наличием непараллелизуемых накладных расходов, связанных с подготовкой данных и оптимизации сетки. Ожидается, что разработчики пакета улучшат эффективность распараллеливания в следующих версиях. Параллельно была проверена работоспособность пакета на СК «Ломоносов».

### 5. Параллельные расчеты обтекания ЛА Х-33

Дополнительно, в том числе для демонстрации корректности построенных тетраэдральных сеток, приведем результаты математического моделирования обтекания ЛА Х-33 с использованием ранее разработанного параллельного пакета внешнего обтекания на основе решения уравнений Навье-Стокса вязкого сжимаемого теплопроводного газа [8], верифицированного с другими авторскими кодами сверхзвукового обтекания аэрокосмических ЛА. Моделирование обтекания ЛА является конечным шагом в технологической цепочке: создание геометрии объекта, генерация неструктурированной тетраэдральной сетки, моделирование обтекания объекта (в том числе суперкомпьютерное). При разработке параллельного пакета внешнего обтекания использовался неинвазивный подход, в котором при распараллеливании вычислительная часть алгоритма для неструктурированных сеток не подвергалась модификации, а все дополнения, связанные с межпроцессорными обменами и синхронизацией, выполнялись до или после вычислительной части кода.

Моделирование обтекания проводилось при числе Маха набегающего потока 6 на высоте  $H=30$  км и углах атаки  $AoA=0, 20$  и  $40$  градусов. Расчет проводился на основе уравнений Навье-Стокса по схеме С.К. Годунова второго порядка точности с использованием глобального ша-

га по времени. Расчет проводился до установления «стационарного» режима обтекания. Моделирование обтекания проводилось на СК «Ломоносов» для сетки в 30 млн. узлов с распараллеливанием на 8 процессоров. Каждый расчет занимал до 1 часа времени.



**Рис. 6.** Распределение давления на поверхности ЛА X-33 при обтекании потоком на высоте 30 км с числом Маха 6 и углами атаки 0, 20 и 40 градусов

На рис. 6 представлены распределения давления, отнесенного к величине давления на бесконечности, на поверхности обтекаемого аппарата при скорости набегающего потока с числом Маха 6 и углах атаки 0, 20 и 40 градусов. Распределение давления демонстрирует пятно повышенного давления в лобовой части ЛА в области его встречи с набегающим потоком, который естественно смещается к нижней плоскости аппарата при увеличении угла атаки. При этом, и растет давление на нижней плоскости аппарата, что вызывает рост как подъемной силы, так и сопротивления. Изменения ударно-волновой структуры потока при изменении угла атаки частично видны на заднем горизонтальном обтекателе, выражаясь в различном угле отхода ударных волн.

## 6. Выводы

Построение пространственных неструктурированных тетраэдральных сеток возможно в полностью автоматическом режиме. Это крайне важно для построения сеток для оперативных расчетов обтекания аэрокосмических и других объектов, обладающих сложной геометрией. В связи с курсом на импортозамещения для промышленности и развития новых технологий исследовался один из наиболее продвинутых открытых open-source пакетов генерации неструктурированных тетраэдральных сеток Gmsh. С помощью пакета удалось успешно и эффективно строить сетки для сложных моделей размерности вплоть до миллиарда ячеек в разумное время на ПК. Были построены сетки для моделей Apollo, X-33 и Space Shuttle. Тщательно исследованы все алгоритмы построения сеток, адаптации и оптимизации и даны рекомендации по их использованию. Исследовано качество генерируемых сеток и ускорение генерации сеток за счет многоядерных вычислений OpenMP. Проведено суперкомпьютерное моделирование обтекания ЛА X-33 на сгенерированных сетках на СК «Ломоносов» при различных углах атаки. Из ранее

известных автору открытых пакетов генерации сеток Gmsh единственный, кто позволяет строить полноценную пространственную сетку с числом ячеек до 1 миллиарда.

Работа выполнена по теме государственного задания (№ госрегистрации АААА-А20-12001169001135-5).

Работа выполнена с использованием ресурсов суперкомпьютерного комплекса МГУ имени М.В. Ломоносова [9].

## Литература

1. Василевский В.Ю., Данилов А.А., Липников К.Н., Чугунов В.Н. Автоматизированные технологии построения неструктурированных расчетных сеток. М: Физматлит, 2016. 216 с.
2. Geuzaine G., Remacle J.F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Engineering* 79(11), 1309-1331, 2009.
3. Marot C., Pellerin J., Remacle J.F, One machine, one minute, three billion tetrahedral. *Int. J. Numer. Methods Engineering* 117(9), 967-990, 2019.
4. <http://gmsh.info>
5. Si H. (2015). TetGen, a Delaunay-based Tetrahedral Mesh Generator. *ACM Transactions on Mathematical Software*. 41 (2), 11:1-11:36, 2015.
6. Si H. Tetgen a quality tetrahedral mesh generator and three-dimensional delaunay triangulator. <http://tetgen.berlios.de/>
7. J. Schöberl. Netgen, an advancing front 2d/3d-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997.
8. Ermakov M.K., Kryukov I.A. Supercomputer modeling of flow past hypersonic flight vehicles. *Journal of Physics: Conference Series* 815(1), 012016, 2017.
9. Воеводин Вл.В., Жуматий С.А., Соболев С.И., Антонов А.С., Брызгалов П.А., Никитенко Д.А., Стефанов К.С., Воеводин Вад.В. Практика суперкомпьютера "Ломоносов" // Открытые системы. – М.: Издательский дом "Открытые системы", N 7, 2012. С. 36-39.

# Исследование масштабируемости апекс-метода для решения сверхбольших задач линейного программирования на кластерных вычислительных системах\*

Л.Б. Соколинский, И.М. Соколинская

Южно-Уральский государственный университет

Статья посвящена исследованию масштабируемости нового метода решения сверхбольших задач линейного программирования. Указанный метод получил название «апекс-метод». Апекс-метод работает по схеме предиктор-корректор. На фазе предиктор находится точка, лежащая на границе  $n$ -мерного многогранника, задающего допустимую область задачи линейного программирования. На фазе корректор организуется итерационный процесс, в результате которого строится последовательность точек, сходящаяся к точному решению задачи линейного программирования. В статье дается формальное описание апекс-метода и приводятся сведения о его параллельной реализации на языке C++ с использованием библиотеки MPI. Приводятся результаты масштабных вычислительных экспериментов на кластерной вычислительной системе по исследованию масштабируемости апекс-метода.

*Ключевые слова:* линейное программирование, задача ЛП большой размерности, апекс-метод, схема предиктор-корректор, итерационный метод, параллельный алгоритм, кластерная вычислительная система.

## 1. Введение

Быстрое развитие технологий накопления и обработки больших данных [1,2] привело к появлению оптимизационных математических моделей в виде сверхбольших задач линейного программирования (ЛП) [3]. Такие задачи возникают в промышленности, экономике, логистике, статистике, квантовой физике и других областях [4–7]. Классическое программное обеспечение во многих случаях не позволяет решить подобные масштабные задачи линейного программирования за приемлемое время [8]. Вместе с тем в ближайшие 2-3 года появятся вычислительные системы экзафлопсного уровня производительности [9], потенциально способные решать подобные задачи. В соответствии с этим актуальной является задача разработки новых эффективных методов для решения сверхбольших задач линейного программирования с помощью экзамастбных вычислительных систем.

До настоящего времени одним из самых распространенных способов решения задачи ЛП являлся класс алгоритмов, предложенных и разработанных Данцигом на основе симплекс-метода [10]. Симплекс-метод оказался эффективным для решения большого класса задач ЛП. Однако симплекс-метод имеет некоторые фундаментальные особенности, ограничивающие его применение для больших задач линейного программирования. Во-первых, в определенных случаях симплекс-методу приходится перебирать все вершины симплекса, что соответствует экспоненциальной временной сложности [11–13]. Во-вторых, симплекс-метод в большинстве случаев удовлетворительно решает задачи ЛП, содержащие до 50000 переменных, однако на больших задачах часто наблюдается потеря точности [14], которая не может быть компенсирована даже путем применения таких ресурсоемких процедур, как «аффинное масштабирование» или «итерационное уточнение» [15]. В-третьих, в общем случае симплекс метод плохо масштабируется на многопроцессорных системах с распределенной памятью. Были предприняты многочисленные попытки построить масштабируемую реализацию симплекс-метода, однако они не увенчались успехом [16]. Во всех случаях граница масштабируемости

---

\* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-07-00092 а и Министерства науки и высшего образования РФ (государственное задание FENU-2020-0022).

составляла от 16 до 32 процессорных узлов (см., например, [17]). Хачиян в [18] предложил вариацию метода эллипсоидов Шора-Юдина-Немировского [19,20], решающую любую задачу ЛП за полиномиальное время, однако попытки применить этот подход на практике оказались безуспешными, так как в подавляющем большинстве случаев алгоритм Хачияна демонстрировал намного худшую эффективность по сравнению с симплекс-методом. Основываясь на работе Хачияна, Кармаркара в работе [21] предложил метод внутренних точек, который демонстрирует полиномиальное время решения задачи ЛП и применим на практике. Итерационные алгоритмы, основанные на методе Кармаркара, способны решать сверхбольшие задачи ЛП с миллионами переменных и миллионами уравнений [22–26]. Более того, эти алгоритмы являются самокорректирующимися, и поэтому обеспечивают высокую точность вычислений. В качестве недостатка метода внутренних точек следует отметить тот факт, что для начала работы алгоритма необходимо иметь точку, удовлетворяющую всем ограничениям задачи линейного программирования. Нахождение такой внутренней точки может сводиться к решению дополнительной задачи линейного программирования [27]. В качестве альтернативы можно указать метод псевдопроекции, основанный на использовании фейеровских отображений [28]. Еще одним существенным недостатком метода внутренних точек является его плохая масштабируемость применительно к многопроцессорным системам с распределенной памятью. Было сделано несколько попыток построить параллельную реализацию для частных случаев (см., например, [29,30]), но эффективную параллельную реализацию для экзаменационных многопроцессорных систем в общем случае построить не удалось. В соответствии с этим является актуальным направление исследований, связанное с поиском новых масштабируемых методов решения задач линейного программирования.

Авторами в работе [3] была предложена идея масштабируемого итерационного метода по схеме предиктор-корректор для решения больших задач линейного программирования, ориентированный на кластерные вычислительные системы. Метод состоит из двух фаз: *Quest* (поиск) и *Target* (позиционирование). На фазе *Quest* происходит поиск допустимой точки, удовлетворяющей системе неравенств, задающих ограничения задачи линейного программирования. На фазе *Target* строится последовательность допустимых точек, сходящаяся к точному решению задачи линейного программирования. Исследованию фазы *Quest* были посвящены работы [3,31–33]. В работе [31] было введено понятие псевдопроекции на выпуклое замкнутое множество, обобщающее понятие проекции. Метод псевдопроекции применяется на фазе *Quest* для нахождения начальной допустимой точки. Для вычисления псевдопроекции используются фейеровские приближения [34], способные «самоисправляться» при накоплении погрешности вычислений. В статье [32] было показано, что при вычислении псевдопроекции на многогранники большой размерности могут эффективно использоваться многоядерные ускорители. В [3] была доказана теорема о необходимом условии сходимости итерационного процесса вычисления псевдопроекции в случае, когда исходные данные изменяются в результате параллельного переноса многогранника. В работе [35] был предложен алгоритм для фазы *Target*, в соответствии с которым формируется специальная система точек, имеющая форму  $n$ -мерного осесимметричного креста, которая передвигается в  $n$ -мерном пространстве таким образом, чтобы решение задачи линейного программирования постоянно находилось в  $\varepsilon$ -окрестности центральной точки креста. Однако, вычислительная сложность этого алгоритма характеризуется экспоненциальной зависимостью от размерности задачи.

В данной статье предлагается и исследуется новый масштабируемый итерационный метод решения больших задач линейного программирования, ориентированный на кластерные вычислительные системы. Указанный метод получил название «апекс-метод». Апекс-метод также состоит из двух фаз – *Quest* и *Target*, однако, в отличие от предыдущего подхода, имеет полиномиальную сложность. Статья организована следующим образом. В разделе 2 дается математическая постановка задачи и формальное описание апекс-метода. В разделе 3 приводятся сведения о параллельной программной реализации фазы *Target* и описываются результаты масштабных вычислительных экспериментов по исследованию масштабируемости апекс-метода на большой кластерной вычислительной системе. В разделе 4 суммируются полученные результаты и намечаются направления дальнейших исследований.

## 2. Апекс-метод

Пусть в пространстве  $\mathbb{R}^n$  задана задача ЛП

$$\bar{x} = \arg \max \{ \langle c, x \rangle \mid Ax \leq b \}^1, \quad (1)$$

где матрица  $A$  имеет  $m$  строк. Мы здесь предполагаем, что ограничение  $x \geq 0$  также включено в систему  $Ax \leq b$  в виде неравенств

$$\begin{array}{cccccccccc} -x_1 & + & 0 & + & \dots & \dots & \dots & + & 0 & \leq & 0; \\ 0 & - & x_2 & + & 0 & + & \dots & + & 0 & \leq & 0; \\ \dots & \dots \\ 0 & + & \dots & \dots & \dots & + & 0 & - & x_n & \leq & 0. \end{array}$$

Обозначим через  $a_i$   $i$ -тую строку матрицы  $A$ . Везде далее мы предполагаем, что  $a_i$  не равна нулевому вектору для всех  $i=1, \dots, m$ . Обозначим через  $M$   $n$ -мерный многогранник, ограничивающий множество допустимых точек задачи (1). Такой многогранник всегда является выпуклым замкнутым множеством. Мы также будем предполагать, что многогранник  $M$  является ограниченным. По определению, точка  $x$  является *граничной* по отношению к многограннику  $M$ , если любая ее окрестность имеет непустое пересечение как с  $M$ , так и с его дополнением. Определим *границу*  $\Gamma_M$  многогранника  $M$  как множество всех его граничных точек.

*Апекс-метод* строится по схеме предиктор-корректор [36] и состоит из двух фаз: *Quest* (предиктор) и *Target* (корректор). Фаза *Quest* находит некоторую точку  $\tilde{x} \in M$ . Фаза *Target*, используя  $\tilde{x}$ , вычисляет последовательность точек  $\{u_0, u_1, \dots, u_k, \dots\}$ , обладающую следующими свойствами:

$$u_k \in \Gamma_M; \quad (2)$$

$$\langle c, u_k \rangle < \langle c, u_{k+1} \rangle; \quad (3)$$

$$\lim_{k \rightarrow \infty} \langle c, u_k \rangle = \bar{x}. \quad (4)$$

Условие (2) означает, что все точки последовательности «лежат» на границе многогранника  $M$ . Условие (3) означает, что значение целевой функции в каждой следующей точке должно быть больше, чем в предыдущей. Условие (4) означает, что последовательность сходится к точному решению задачи (1).

Фаза *Quest* находит точку  $\tilde{x} \in M$ , используя операцию *псевдопроектирования* [28], являющуюся обобщением операции ортогонального проектирования. Дадим ее формальное определение. Пусть задано отображение  $\varphi_M : \mathbb{R}^n \rightarrow \mathbb{R}^n$ :

$$\varphi_M(x) = \frac{1}{h} \sum_{i=1}^m \rho_i^+(x), \quad (5)$$

где

$$\rho_i^+(x) = \frac{\max \{ \langle a_i, x \rangle - b_i, 0 \}}{\|a_i\|^2} a_i, \quad (6)$$

$h$  – количество ненулевых слагаемых в сумме  $\sum_{i=1}^m \rho_i^+(x)$ . Тогда *псевдопроекция*  $\pi_M(x)$  точки  $x$  на многогранник  $M$  определяется следующей формулой:

$$\pi_M(x) = \lim_{k \rightarrow \infty} \varphi_M^{(k)}(x), \quad (7)$$

<sup>1</sup>  $\langle c, x \rangle$  обозначает скалярное произведение двух векторов.

---

**Алгоритм 1.**

---

```

1: input  $x_0$ 
2:  $k := 0$ 
3:  $x_{k+1} := x_{k+1} - \varphi_M(x_k)$ 
4: if  $\|x_{k+1} - x_k\| < \varepsilon$  go to 7
5:  $k := k + 1$ 
6: go to 3
7: output  $\tilde{x} = x_{k+1}$ 
8: stop

```

---

**Рис. 1.** Алгоритм вычисления псевдопроекции.

где

$$\varphi_M^{(k)}(x) = \underbrace{\varphi_M(\varphi_M(\dots\varphi_M(x)\dots))}_k.$$

Итерационный алгоритм, реализующий операцию псевдопроектирования, приведен на рис. 1. Алгоритм завершает свою работу, когда расстояние между соседними приближениями становится меньше малой величины  $\varepsilon > 0$ , являющейся параметром алгоритма. Доказательство сходимости алгоритма 1 можно найти в статье [37]. В работе [38] авторами была предложена и исследована масштабируемая параллельная реализация алгоритма 1 в виде операций над списками. С использованием стоимостной метрики модели параллельных вычислений BSF [39] было показано, что граница масштабируемости<sup>1</sup> указанного параллельного алгоритма может быть оценена, как  $O(\sqrt{n})$ .

Фаза *Target* вычисляет последовательность точек  $\{u_0, u_1, \dots, u_k, \dots\}$ , удовлетворяющую свойствам (2) – (4), сходящуюся к точному решению задачи (1). Для вычисления начального приближения  $u_0$  используется точка *апекса*  $z$ , которая определяется следующим образом. Пусть  $\tilde{x} \in M$  – точка, полученная на фазе *Quest*. Обозначим через  $S$   $n$ -мерный шар с радиусом  $r$  и центром в точке  $\tilde{x}$ , обладающий свойством:  $M \subset S$  (см. рис. 2). Зафиксируем некоторое положительное число  $\sigma \in \mathbb{R}_{>0}$ , такое, что  $\sigma \gg r$ . Определим единичный вектор

$$e_c = \frac{c}{\|c\|},$$

где  $c$  – вектор коэффициентов целевой функции задачи (1). Тогда *точка апекса* вычисляется по формуле

$$z = \tilde{x} + \sigma e_c. \quad (8)$$

Зададим начальное приближение  $u_0$  следующим образом:

$$u_0 = \pi_M(z),$$

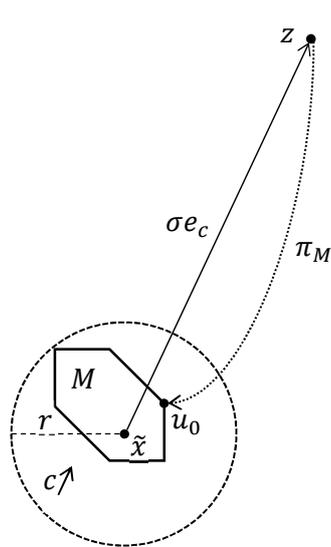
то есть  $u_0$  получается в результате псевдопроектирования точки *апекса*  $z$  на многогранник  $M$ . Используя схему доказательства сходимости алгоритма из [37], несложно показать, что в этом случае точка  $u_0$  будет лежать на границе  $\Gamma_M$  многогранника  $M$ . Это означает, что найдется гиперплоскость  $H_i$  ( $i \in \{1, \dots, m\}$ ), определяемая уравнением  $\langle a_i, x \rangle = b_i$ , такая, что

$$u_0 \in H_i \cap \Gamma_M. \quad (9)$$

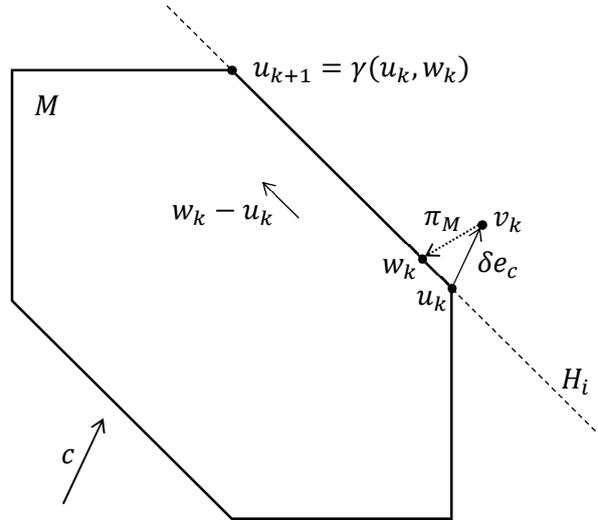
Таким образом для точки  $u_0$  выполняется условие (2).

---

<sup>1</sup> Под границей масштабируемости параллельного алгоритма для кластерной вычислительной системы понимается максимальное количество вычислительных узлов, вплоть до которого наблюдается рост ускорения, то есть распараллеливание является эффективным.



**Рис. 2.** Построение точки апекса  $z$  и начального приближения  $u_0$ .



**Рис. 3.** Построение точки  $w_k$  и следующего приближения  $u_{k+1} = \gamma(u_k, w_k)$ .

Теперь предположим, что уже найдено приближение  $u_k$  ( $k \in \mathbb{Z}_{\geq 0}$ ), удовлетворяющее условиям (2) и (3). Для построения следующего приближения  $u_{k+1}$  вычислим промежуточную точку

$$v_k = u_k + \delta e_c, \quad (10)$$

которая получается путем прибавления к точке  $u_k$  единичного вектора  $e_c$ , сонаправленного с вектором коэффициентов целевой функции  $c$ , умноженным на малую величину  $\delta \in \mathbb{R}_{>0}$  (см. рис. 3). Применив операцию псевдопроектирования  $\pi_M$  к точке  $v_k$ , получаем следующую промежуточную точку

$$w_k = \pi_M(v_k). \quad (11)$$

Поскольку отображение  $\varphi_M$ , используемое для вычисления псевдопроекции в формуле (7), является однозначным  $M$ -фейеровским<sup>1</sup> отображением [34], то при достаточно малом  $\delta$  из (9) и (10) следует, что  $w_k \in H_i \cap \Gamma_M$ , то есть  $w_k$  будет лежать на той же грани, что и  $u_k$ . Если значение целевой функции в точке  $u_k$  будет больше либо равно значения целевой функции в точке  $w_k$ , то точка  $u_k$  принимается за приближенное решение задачи (1). Предположим, что справедливо обратное, то есть  $\langle c, w_k \rangle > \langle c, u_k \rangle$ . Определим луч  $L_{u_k w_k}$ , исходящий из точки  $u_k$  в направлении точки  $w_k$ :

$$L_{u_k w_k} = \{x \in \mathbb{R}^n \mid x = u_k + \eta(w_k - u_k), \eta \in \mathbb{R}_{\geq 0}\}.$$

Определим отображение  $\gamma: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  следующим образом:

$$\gamma(u_k, w_k) = \arg \max \{\|x - u_k\| \mid x \in L_{u_k w_k} \cap M\}. \quad (12)$$

Другими словами, отображение  $\gamma$  вычисляет точку, лежащую на луче  $L_{u_k w_k}$ , принадлежащую многограннику  $M$ , и максимально удаленную от точки  $u_k$ . Указанную точку возьмем в качестве следующего приближения:

$$u_{k+1} = \gamma(u_k, w_k).$$

Очевидно, что для точки  $u_{k+1}$  также будут выполняться условия (2) и (3).

<sup>1</sup> Однозначное отображение  $\varphi_M: \mathbb{R}^n \rightarrow \mathbb{R}^n$  называется фейеровским относительно множества  $M$  или кратко  $M$ -фейеровским, если  $\forall y \in M (\varphi(y) = y) \vee \forall x \notin M (\forall y \in M (\|\varphi(x) - y\| < \|x - y\|))$ .

---

**Алгоритм 2.**

---

```

1: input  $\tilde{x}$ 
2:  $e_c := c/\|c\|$ 
3:  $z := \tilde{x} + \sigma e_c$ 
4:  $u_0 := \pi_M(z)$ 
5:  $k := 0$ 
6:  $v_k := u_k + \delta e_c$ 
7:  $w_k := \pi_M(v_k)$ 
8: if  $\langle c, w_k \rangle \leq \langle c, u_k \rangle$  go to 12
9:  $u_{k+1} = \gamma(u_k, w_k)$ 
10:  $k := k + 1$ 
11: go to 6
12: output  $\bar{x} = u_k$ 
13: stop

```

---

Рис. 4. Реализация фазы Target.

---

**Алгоритм 3.**

---

```

1: input  $u_k, w_k$ 
2:  $e_{u_k w_k} := (w_k - u_k)/\|w_k - u_k\|$ 
3:  $\tau_0 := \mu; t_0 := u_k$ 
4:  $j := 0; i := 0$ 
5: if  $\tau_i < \varepsilon$  go to 13
6:  $t_{j+1} := t_j + \tau_i e_{u_k w_k}$ 
7: if  $t_{j+1} \notin M$  go to 10
8:  $j := j + 1$ 
9: go to 6
10:  $\tau_{i+1} := \tau_i/2$ 
11:  $i := i + 1$ 
12: go to 5
13: output  $u_{k+1} = t_j; \text{stop}$ 

```

---

Рис. 5. Алгоритм вычисления отображения  $\gamma$ .

Алгоритм, реализующий фазу Target приведен на рис. 4. На шаге 1 вводится точка  $\tilde{x} \in M$ , полученная на фазе Quest с помощью алгоритма 1. На шаге 2 вычисляется единичный вектор  $e_c$ , сонаправленный с вектором  $c$ , координаты которого совпадают с коэффициентами целевой функции задачи (1). На шаге 3 вычисляется точка апекса  $z$  в соответствии с формулой (8) (см. рис. 2). На шаге 4 с помощью алгоритма 1 вычисляется начальное приближение  $u_0$ , представляющее собой псевдопроекцию точки апекса  $z$  на многогранник  $M$  (см. формулу (7) и рис. 2). На шаге 5 счетчик итераций  $k$  устанавливается в начальное значение 0. Итерационный цикл начинается на шаге 6, который вычисляет промежуточную точку  $v_k$  по формуле (10). На шаге 7 по формуле (11) вычисляется промежуточная точка  $w_k$ , лежащая на той же грани многогранника  $M$ , что и текущее приближение  $u_k$  (см. рис. 3). На шаге 8 проверяется условие завершения: если значение целевой функции в точке  $w_k$  не превышает значения целевой функции в точке  $u_k$ , то итерационный процесс заканчивается, и  $u_k$  выводится в качестве приближенного решения задачи (1). В противном случае точка  $w_k$  задает направление  $(w_k - u_k)$ , в котором значение целевой функции будет увеличиваться. На шаге 9 вычисляется допустимая точка  $u_{k+1}$ , в которой достигается максимум целевой функции по направлению  $(w_k - u_k)$ . Шаг 10 увеличивает на единицу счетчик итераций  $k$ . На шаге 11 осуществляется переход на шаг 6, начинающий следующую итерацию. Сходимость алгоритма 2 непосредственно следует из замкнутости и ограниченности многогранника  $M$ . Таким образом, условие (4) выполняется.

Алгоритм вычисления приближенного значения отображения  $\gamma$ , определяемого формулой (12), и используемого на шаге 9 алгоритма 2, приведен на рис. 5. Указанный алгоритм вычисляет последовательность точек  $\{t_0, t_1, \dots, t_j, \dots\}$ , такую, что

$$t_0 = u_k, \quad (13)$$

$$\lim_{j \rightarrow \infty} t_j = \gamma(u_k, w_k). \quad (14)$$

На шаге 1 вводятся исходные значения  $u_k$  и  $w_k$ . На шаге 2 вычисляется единичный вектор  $e_{u_k w_k}$ , сонаправленный с вектором  $(w_k - u_k)$ . На шаге 3 в качестве начального значения приращения  $\tau_0$  устанавливается константа  $\mu \in \mathbb{R}_{>0}$ , являющаяся параметром алгоритма, а в качестве начального приближения  $t_0$  устанавливается точка  $u_k$ . На шаге 4 счетчики итераций  $j, i$

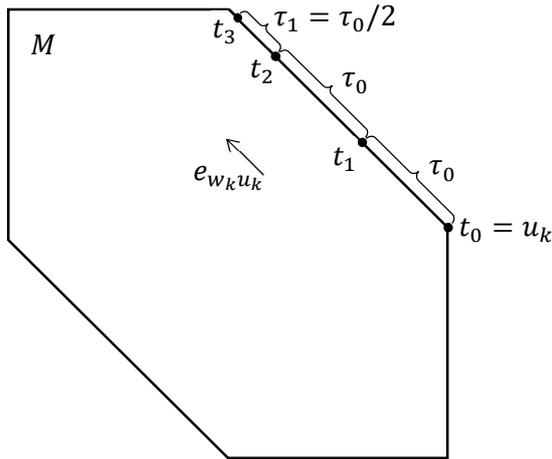


Рис. 6. Иллюстрация работы алгоритма 3.

$$\begin{cases} x_0 & \leq 200 \\ x_1 & \leq 200 \\ \dots & \dots \\ x_{n-1} & \leq 200 \\ x_0 + x_1 + \dots + x_{n-1} & \leq 200(n-1) + 100 \\ x_0 + x_1 + \dots + x_{n-1} & \geq 100 \\ x_0 & \geq 0 \\ x_1 & \geq 0 \\ \dots & \dots \\ x_{n-1} & \geq 0 \end{cases}$$

Рис. 7. Масштабируемая система неравенств.

устанавливаются в значение 0. На шаге 5 проверяется условие завершения: если приращение  $\tau_i$  меньше малой величины  $\varepsilon \in \mathbb{R}_{>0}$ , являющейся параметром алгоритма, то происходит переход на шаг 13, где в качестве результата выводится  $u_{k+1} = t_j$ , после чего алгоритм завершает свою работу. В противном случае выполняется шаг 6, вычисляющий следующее приближение  $t_{j+1}$  путем смещения точки  $t_j$  в направлении  $(w_k - u_k)$  на расстояние  $\tau_i$ . На шаге 7 проверяется, не вышла ли точка  $t_{j+1}$  за пределы  $M$ . Если это имеет место, то происходит переход на шаг 10, где в качестве новой величины смещения  $\tau_{i+1}$  устанавливается  $\frac{1}{2}\tau_i$ . На шаге 11 счетчик  $i$  увеличивается на единицу и происходит переход на шаг 5 для повторного вычисления точки  $t_{j+1}$  с использованием меньшего смещения  $\tau_i$ . Если при проверке условия на шаге 7 оказывается, что новое приближение  $t_{j+1}$  принадлежит многограннику  $M$ , то выполняется шаг 8, увеличивающий на единицу счетчик  $j$ , и происходит переход на шаг 6 (см. рис. 6). Очевидно, что последовательность точек  $\{t_0, t_1, \dots, t_j, \dots\}$ , генерируемая алгоритмом 3, удовлетворяет условиям (13), (14).

### 3. Программная реализация и вычислительные эксперименты

Нами была выполнена параллельная реализация алгоритма 2 на языке C++ с использованием программного каркаса BSF [40,41] и библиотеки параллельного программирования MPI. Схема параллельной реализации идентична схеме, описанной в статье [42]. Исходные коды свободно доступны в сети Интернет по адресу <https://github.com/leonid-sokolinsky/Apex-method>. В качестве задачи была использована масштабируемая система неравенств размерности  $n$  из статьи [32] (см. рис. 7). Количество неравенств  $m$  в этой системе вычисляется по формуле  $m = 2n + 2$ . В качестве вектора коэффициентов целевой функции был взят вектор

$$c = (10n, 10(n-1), 10(n-2), \dots, 10). \tag{15}$$

В этом случае точным решением задачи (1) для любой размерности  $n \geq 2$  будет являться точка

$$\bar{x} = (200, 200, \dots, 200, 100).$$

Вычислительные эксперименты проводились на вычислительном кластере «Торнадо ЮУрГУ» [43], характеристики которого приведены на рис. 8. Результаты экспериментов представлены на рис. 9. Вычисления проводились для размерностей 5 000, 7 500 и 10 000. Количество неравенств, соответственно, составляло 10 002, 15 002 и 20 002. Эксперименты показали, что граница масштабируемости апекс-метода существенным образом зависит от размера задачи. При  $n = 5000$  граница масштабируемости составила приблизительно 55

Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores 3.33 GHz)
Количество процессоров в узле	2
Оперативная память узла	24 GB DDR3
Соединительная сеть	InfniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

Рис. 8. Характеристики вычислительного кластера «Торнадо ЮУрГУ».

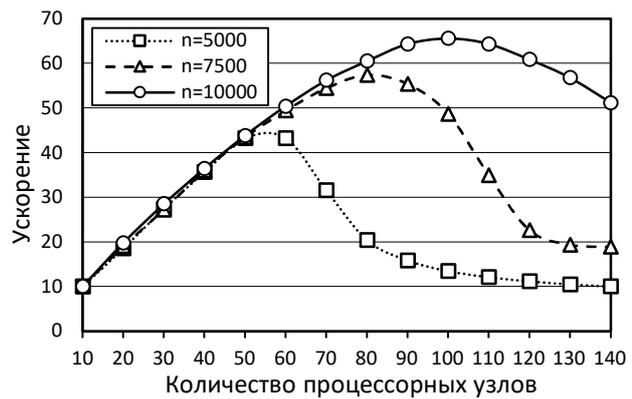


Рис. 9. Эксперименты по исследованию масштабируемости алгоритма 2.

процессорных узлов. Для задачи размерности  $n = 7500$  эта граница увеличилась до 80 узлов, а на задаче размерности  $n = 10000$  она достигла 100 узлов. Дальнейшее увеличение размерности задачи приводило к нехватке оперативной памяти на процессорных узлах. Следует отметить, что вычисления производились с двойной точностью, при которой вещественное число представляется в формате с плавающей точкой и занимает 64 разряда. Попытка перейти на одинарную точность, требующую только 32 разряда для представления вещественного числа, оказалась неудачной из-за операции вычисления псевдопроекции, используемой на шаге 4 алгоритма 2. В этом случае алгоритм 1, вычисляющий псевдопроекцию, переставал сходиться. Эксперименты также показали, что параметр  $\sigma$ , определяющий в соответствии с формулой (8) удаленность точки апекса  $z$  от многогранника  $M$  (см. рис. 2), при больших значениях мало влияет на общее время решения задачи. Для модельного примера на рис. 7 и целевой функции с коэффициентами из формулы (15) параметр  $\sigma$  не может быть меньше  $200n$ , так как в этом случае точка апекса  $z$  окажется внутри многогранника  $M$ . Если точка апекса находится на недостаточно большом расстоянии от многогранника, то ее псевдопроекция оказывается внутренней точкой некоторой грани. Если же точку апекса взять достаточно далеко (в экспериментах использовалось значение  $\sigma = 20000n$ ), то псевдопроекция в исследуемом примере всегда попадала в одну из вершин многогранника. Также отметим, что во всех случаях приближения  $u_0, u_1, u_2, \dots$  оказывались вершинами многогранника. Вычислительный эксперимент показал, что более 99% времени решения задачи ЛП апекс-методом приходится на вычисление псевдопроекций (шаг 4 алгоритма 2). При этом вычисление одного приближения  $u_k$  для задачи размерности  $n = 10000$  на 100 процессорных узлах составило 44 минуты.

## 4. Заключение

В статье предложен новый масштабируемый итерационный метод решения задачи линейного программирования (ЛП), получивший название «апекс-метод». Апекс-метод строится по схеме предиктор-корректор и состоит из двух фаз. На первой фазе, называемой *Quest*, происходит поиск допустимой точки задачи ЛП с использованием метода псевдопроекций. На второй фазе, называемой *Target*, вычисляется последовательность точек на поверхности многогранника, ограничивающего допустимую область, сходящаяся к точному решению задачи ЛП. Указанный метод реализован в виде программы на языке C++ с использованием библиотеки параллельного программирования MPI. Описаны вычислительные эксперименты по решению больших задач ЛП на кластерной вычислительной системе. Проведенные эксперименты показали, что апекс-метод масштабируется с ростом размерности задачи. Сильной стороной апекс-метода является его «самокорректируемость» по отношению к возникающим погрешностям вычислений. Апекс-метод также потенциально может использоваться для решения нестационарных задач ЛП. Недостатком метода является высокая вычислительная сложность построения псевдопроекций.

В качестве направлений дальнейших исследований предполагаются следующие.

1. Выполнить подробное математическое доказательство сходимости апекс-метода.
2. Выполнить тестирование апекс-метода на случайных задачах ЛП, генерируемых специальным алгоритмом.
3. Решить апекс-методом выборочные задачи ЛП из репозитория Netlib-Lp [44,45].
4. Использовать апекс-метод для генерации тестового набора данных для разработки и обучения сверточных нейронных сетей, способных в кооперации с суперкомпьютером быстро решать сверхбольшие задачи ЛП.

## Литература

1. Jagadish H. V. et al. Big data and its technical challenges // Communications of the ACM. ACM, 2014. Vol. 57, no. 7. P. 86–94. DOI:10.1145/2611567.
2. Hartung T. Making Big Sense From Big Data // Frontiers in Big Data. 2018. Vol. 1. P. 5. DOI:10.3389/fdata.2018.00005.
3. Соколинская И.М., Соколинский Л.Б. О решении задачи линейного программирования в эпоху больших данных // Параллельные вычислительные технологии – XI международная конференция, ПаВТ'2017, г. Казань, 3–7 апреля 2017 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2017. С. 471–484.
4. Chung W. Applying large-scale linear programming in business analytics // 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2015. P. 1860–1864. DOI:10.1109/IEEM.2015.7385970.
5. Gondzio J. et al. Solving large-scale optimization problems related to Bell's Theorem // Journal of Computational and Applied Mathematics. Elsevier B.V., 2014. Vol. 263. P. 392–404. DOI:10.1016/j.cam.2013.12.003.
6. Sodhi M.S. LP modeling for asset-liability management: A survey of choices and simplifications // Operations Research. 2005. Vol. 53, no. 2. P. 181–196. DOI:10.1287/opre.1040.0185.
7. Brogaard J., Hendershott T., Riordan R. High-Frequency Trading and Price Discovery // Review of Financial Studies. 2014. Vol. 27, no. 8. P. 2267–2306. DOI:10.1093/rfs/hhu032.
8. Bixby R.E. Solving Real-World Linear Programs: A Decade and More of Progress // Operations Research. INFORMS, 2002. Vol. 50, no. 1. P. 3–15. DOI:10.1287/opre.50.1.3.17780.
9. Dongarra J., Gottlieb S., Kramer W.T.C. Race to Exascale // Computing in Science & Engineering. 2019. Vol. 21, no. 1. P. 4–5. DOI:10.1109/MCSE.2018.2882574.
10. Dantzig G.B. Linear programming and extensions. Princeton, N.J.: Princeton university press, 1998. 656 p.
11. Klee V., Minty G.J. How good is the simplex algorithm? // Inequalities - III. Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, Sept. 1-9, 1969 / ed. Shisha O. New York-London: Academic Press, 1972. P. 159–175.
12. Jeroslow R.G. The simplex algorithm with the pivot rule of maximizing criterion improvement // Discrete Mathematics. 1973. Vol. 4, no. 4. P. 367–377. DOI:10.1016/0012-365X(73)90171-4.
13. Zadeh N. A bad network problem for the simplex method and other minimum cost flow algorithms // Mathematical Programming. Springer, 1973. Vol. 5, no. 1. P. 255–266. DOI:10.1007/BF01580132.
14. Bartels R.H., Stoer J., Zenger C. A Realization of the Simplex Method Based on Triangular Decompositions // Handbook for Automatic Computation. Volume II: Linear Algebra. Berlin, Heidelberg: Springer, 1971. P. 152–190. DOI:10.1007/978-3-642-86940-2\_11.
15. Tolla P. A Survey of Some Linear Programming Methods // Concepts of Combinatorial Optimization. 2nd ed. / ed. Paschos V.T. Hoboken, NJ, USA: John Wiley & Sons, 2014. P. 157–188. DOI:10.1002/9781119005216.ch7.
16. Mamalis B., Pantziou G. Advances in the Parallelization of the Simplex Method // Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science, vol. 9295 / ed. Zaroliagis C., Pantziou G., Kontogiannis S. Cham: Springer, 2015. P. 281–307. DOI:10.1007/978-3-319-24024-4\_17.

17. Lubin M. et al. Parallel distributed-memory simplex for large-scale stochastic LP problems // Computational Optimization and Applications. 2013. Vol. 55, no. 3. P. 571–596. DOI:10.1007/s10589-013-9542-y.
18. Хачиян Л.Г. Полиномиальный алгоритм в линейном программировании // Доклады Академии наук СССР. 1979. Т. 244, № 5. С. 1093–1096.
19. Шор Н.З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования // Кибернетика. 1977. № 1. С. 94–95.
20. Юдин Д.Б., Немировский А.С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач // Экономика и математические методы. 1976. № 2. С. 357–369.
21. Karmarkar N. A new polynomial-time algorithm for linear programming // Combinatorica. 1984. Vol. 4, no. 4. P. 373–395. DOI:10.1007/BF02579150.
22. Fathi-Hafshejani S. et al. Primal–dual interior-point method for linear optimization based on a kernel function with trigonometric growth term // Optimization. 2018. Vol. 67, no. 10. P. 1605–1630. DOI:10.1080/02331934.2018.1482297.
23. Asadi S., Mansouri H. A Mehrotra type predictor-corrector interior-point algorithm for linear programming // Numerical Algebra, Control and Optimization. American Institute of Mathematical Sciences, 2019. Vol. 9, no. 2. P. 147–156. DOI:10.3934/naco.2019011.
24. Yuan Y. Implementation tricks of interior-point methods for large-scale linear programs // Proc. SPIE, vol. 8285. International Conference on Graphic and Image Processing (ICGIP 2011). International Society for Optics and Photonics, 2011. DOI:10.1117/12.913019.
25. Kheirfam B., Haghighi M. A full-Newton step infeasible interior-point method for linear optimization based on a trigonometric kernel function // Optimization. Informa UK Limited, 2016. Vol. 65, no. 4. P. 841–857. DOI:10.1080/02331934.2015.1080255.
26. Xu Y., Zhang L., Zhang J. A full-modified-newton step infeasible interior-point algorithm for linear optimization // Journal of Industrial and Management Optimization. American Institute of Mathematical Sciences, 2016. Vol. 12, no. 1. P. 103–116. DOI:10.3934/jimo.2016.12.103.
27. Roos C., Terlaky T., Vial J.-P. Interior Point Methods for Linear Optimization. New York: Springer, 2005. 500 p. DOI:10.1007/b100325.
28. Sokolinskaya I. Parallel Method of Pseudoprojection for Linear Inequalities // Parallel Computational Technologies. PCT 2018. Communications in Computer and Information Science, vol. 910. Cham: Springer, 2018. P. 216–231. DOI:10.1007/978-3-319-99673-8\_16.
29. Hafsteinsson H., Levkovitz R., Mitra G. Solving large scale linear programming problems using an interior point method on a massively parallel SIMD computer // Parallel Algorithms and Applications. Taylor & Francis Group, 1994. Vol. 4, no. 3–4. P. 301–316. DOI:10.1080/10637199408915470.
30. Karypis G., Gupta A., Kumar V. A parallel formulation of interior point algorithms // Proceedings of the 1994 ACM/IEEE conference on Supercomputing (Supercomputing'94). Los Alamitos, CA, USA: IEEE Computer Society Press, 1994. P. 204–213. DOI:10.1109/SUPER.1994.344280.
31. Ершова А.В., Соколинская И.М. О сходимости масштабируемого алгоритма построения псевдопроекции на выпуклое замкнутое множество // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. 2011. № 37(254). С. 12–21.
32. Соколинская И.М., Соколинский Л.Б. Модифицированный следящий алгоритм для решения нестационарных задач линейного программирования на кластерных вычислительных системах с многоядерными ускорителями // Суперкомпьютерные дни в России: труды международной конференции (26-27 сентября 2016 г., г. Москва). Москва: Изд-во МГУ, 2016. С. 294–306.
33. Соколинская И.М., Соколинский Л.Б. Исследование масштабируемости модифицированного алгоритма Чиммино для линейных неравенств // Суперкомпьютерные дни в России: Труды международной конференции (24-25 сентября 2018 г., г. Москва). Москва: Изд-во МГУ, 2018. С. 673–983.
34. Ерёмин И.И. Фейеровские методы для задач выпуклой и линейной оптимизации. Челябинск: Издательский центр ЮУрГУ, 2009. 199 с.
35. Соколинская И.М., Соколинский Л.Б. Масштабируемый алгоритм для решения нестационарных задач линейного программирования // Вычислительные методы и

- программирование: новые вычислительные технологии. 2018. Т. 19, № 4. С. 540–550. DOI:10.26089/NumMet.v19r448.
36. Press W.H. et al. Numerical Recipes: The Art of Scientific Computing. 3rd ed. Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, Sao Paulo: Cambridge University Press, 2007.
  37. Censor Y. et al. New Methods for Linear Inequalities // SIAM Journal on Scientific Computing. Society for Industrial and Applied Mathematics, 2008. Vol. 30, no. 1. P. 473–504. DOI:10.1137/050639399.
  38. Соколинский Л.Б., Соколинская И.М. Параллельный алгоритм решения нестационарных систем линейных неравенств // Параллельные вычислительные технологии – XIV международная конференция, ПаВТ'2020, г. Пермь, 31 марта–2 апреля 2020 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2020. С. 275–286. DOI:10.14529/pct2020.
  39. Ежова Н.А., Соколинский Л.Б. BSF: модель параллельных вычислений для многопроцессорных систем с распределенной памятью // Параллельные вычислительные технологии – XII международная конференция, ПаВТ'2018, г. Ростов-на-Дону, 2–6 апреля 2018 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2018. С. 253–265.
  40. Ежова Н.А., Соколинский Л.Б. Модель параллельных вычислений для многопроцессорных систем с распределенной памятью // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2018. Т. 7, № 2. С. 32–49. DOI:10.14529/cmse180203.
  41. Sokolinsky L.B. BSF-skeleton [Electronic resource]. URL: <https://github.com/leonid-sokolinsky/BSF-skeleton>. (accessed: 18.06.2019)
  42. Ежова Н.А., Соколинский Л.Б. Модель параллельных вычислений BSF-MR // Системы управления и информационные технологии. 2019. № 3(77). С. 15–21.
  43. Kostenetskiy P.S., Safonov A.Y. SUSU Supercomputer Resources // Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT 2016). CEUR Workshop Proceedings. Vol. 1576. 2016. P. 561–573.
  44. Gay D.M. Netlib-Lp [Electronic resource]. URL: <http://www.netlib.org/lp/> (accessed: 18.06.2019).
  45. Koch T. The final NETLIB-LP results // Operations Research Letters. 2004. Vol. 32, no. 2. P. 138–142. DOI:10.1016/S0167-6377(03)00094-4.

# Многокритериальная оптимизация модели гребного винта с учетом влияния корпуса судна

М.П. Лобачев, В.Н. Новоселов, Т.И. Сайфуллин, А.Е. Таранов

ФГУП «Крыловский государственный научный центр»

В качестве объектов исследования выбраны модель контейнеровоза KCS и его гребной винт. Продемонстрированы возможности современного программного обеспечения, позволяющего выполнять многокритериальную оптимизацию судовых движителей с учетом влияния корпуса судна. В работе приводится анализ геометрических характеристик модели гребного винта, измененных в процессе оптимизации в зависимости от выбранных критериев, результаты численного моделирования контрольных самоходных испытаний, оценка виброакустических и кавитационных характеристик моделей оптимизированных гребных винтов. Полученные решения характеризуются повышенным КПД и снижением пульсаций давления в пространстве, что достигается за счет незначительного снижения запасов на кавитацию.

*Ключевые слова:* численное моделирование, оптимизация, параметрическая модель.

## 1. Введение

Обострение конкурентной борьбы в современной мировой экономике заставляет производителей вести активный поиск путей повышения технических характеристик объектов. В связи с этим, оптимизация на сегодняшний день является одной из самых востребованных задач в промышленности и поэтому широко применяется в таких сферах как авиастроение, космическая техника, автомобилестроение, турбомашиностроение и судостроение. При решении задач оптимизации перед инженерами и исследователями встают такие вопросы как улучшение характеристик конструкции при сохранении прочности, повышение коэффициента полезного действия исследуемого объекта без необходимости увеличения потребной мощности, уменьшение веса или габаритов изделия без потери основных функциональных возможностей и т.д.

В последние несколько лет в Крыловском государственном научном центре ведутся работы по созданию технологии многокритериальной оптимизации судовых движителей с использованием суперкомпьютерной техники и последних достижений в области разработки программного обеспечения. При этом для расчета гидродинамических характеристик как движителя, так и корпуса судна используются методы, основанные на решении уравнений Рейнольдса (RANS). Это позволяет проводить оптимизацию с более корректным учетом влияния вязкости, в том числе и с учетом реального взаимодействия движителя и корпуса судна. По отношению к движителю это дает возможность более точного учета локального влияния вязкости, особенно на корневых и концевых сечениях [1]. Работы последних лет по данной тематике [2] локальный учет вязкости на винте не обеспечивают, но для интегральных гидродинамических характеристик (ГДХ) дают близкие результаты. Предварительные результаты оптимизации моделей гребных винтов в однородном потоке, полученные в Крыловском научном центре, опубликованы в работах [3, 4]. В работе [4] приведен единичный оптимизационный расчет с одним критерием. Полученное решение характеризуется нарушением условий задачи. При заданном строгом ограничении на располагаемый момент на валу, требуемый момент оптимизированного гребного винта превышает его на 5% [4, стр. 103].

Научная новизна настоящей работы заключается в создании нового многокритериального оптимизационного алгоритма, свободного от ошибок, присущих работе [4], и выполнении гидродинамического и кавитационного анализа полученных оптимизированных форм гребных винтов. Под оптимизационным алгоритмом здесь понимается вся совокупность математических методов, используемых для решения задачи оптимизации, а не только собственно алгоритмы оптимизации (генетический, градиентный, суррогатный и т.п.). Его новизна заключается в части доработки методики расчета ГДХ объектов с целью обеспечения возможности исполь-

зования RANS-методов при проведении оптимизации. Особенно в части корректного учета взаимодействия винта и корпуса и влияния вязкости на локальные гидродинамические характеристики. Выбор управляющих параметров в зависимости от заданных критериев поиска производится на основе предварительного корреляционного анализа. Устойчивость оптимизационного алгоритма проверяется путем многократного повторения поиска с различными начальными условиями.

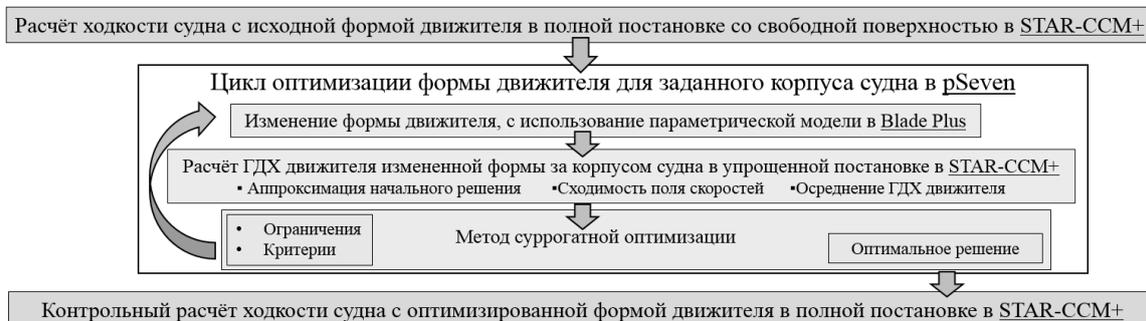


Рис.1. Схема оптимизационного алгоритма.

В качестве объекта исследования также, как и в работе [4], используется международный тестовый объект – контейнеровоз KCS, хорошо изученный методами физического и математического моделирования. Оптимизационная процедура выполняется на режиме, соответствующему тестовой задаче № 2.5 Токийского симпозиума по вычислительной гидродинамике 2015 года [5-7]. Результаты численного моделирования данной тестовой задачи выступают в качестве реперной точки для сравнения с решениями, полученными в ходе оптимизации. Оптимизация выполняется применительно к модельным условиям. На данном этапе не ставилась задача разработки полной технологии проектирования движителя. Проверялась только возможность получения движителя с улучшенными характеристиками в зависимости от критериев оптимизации. Тем более, что данный пример чисто гипотетический и такого объекта KCS как реального судна – не существует.

Процесс оптимизации проходит под управлением российского программного комплекса pSeven [8]. Используется современный оптимизационный алгоритм на основе метамоделей (surrogate-based optimization), который характеризуется низкими требованиями к вычислительным ресурсам. Суть данного алгоритма заключается в замене полной модели, на быстро вычисляемую суррогатную модель. Обзор суррогатных методов оптимизации подробно рассмотрен в работе [9]. В качестве вычислительного ядра используется программа Star-CCM+, а генерация трехмерной параметрической модели гребного винта производится в программе BladePlus, разработанной в Крыловском государственном научном центре ведущим инженером-программистом Новоселовым В.Н. [10] Программа Blade Plus реализует требования отраслевого стандарта «ОСТ 5.0317-80. Винты гребные фиксированного шага морских судов. Правила математического представления поверхности лопастей», дополняя их алгоритмами построения гидродинамически цельных параметрических поверхностей и механизмами трансформации на основе преобразований «расширенных» сплайнов (free form deformation), содержащих как исходное распределение, так и его гладкую деформацию, которая может быть задана не более чем четырьмя точками [11].

В качестве целевых функций при оптимизации использованы максимум коэффициента полезного действия (КПД) гребного винта и минимум пульсаций давления в пространстве около гребного винта. В качестве ограничений используются требуемый упор движителя, располагаемый момент на гребном валу и риски возникновения второй стадии кавитации. Для полученных оптимизированных гребных винтов выполняется проверка требований по прочности и оценка гидродинамических характеристик с учетом кавитационных явлений.

## 2. Постановка задачи определения гидродинамических характеристик судна и движителя. Обоснование выбранных моделей течения

На Рисунке 2 представлен внешний вид модели контейнеровоза KCS. Длина модели составляет 7,27 м. Начало системы координат расположено на пересечении диаметральной плоскости и невозмущенной свободной поверхности, вертикальная ось совпадает с осью баллера руля. Ось  $x$  направлена в нос судна,  $y$  – на левый борт,  $z$  – вверх.



Рис.2. Общий вид модели контейнеровоза KCS.

Геометрическая модель гребного винта KP505, спроектированного в KRISO для контейнеровоза KCS, доступна на сайте Токийского симпозиума 2015 года [12]. Диаметр модели гребного винта составляет 0,25 м. Для выполнения оптимизационной процедуры в программе BladePlus разработана параметрическая модель гребного винта KP505. В связи с ограниченностью объема данной статьи, описание метода параметризации модели гребного винта не приводится. Описание алгоритмов параметризации представлено в работе [3] и в научно-технических отчетах ФГУП «Крыловский государственный научный центр».

Для решения задачи численного моделирования самоходных испытаний использован пакет вычислительной гидродинамики Star-CCM+ фирмы Siemens. Построение гексаэдральных расчетных сеток выполнено с помощью встроенного в пакет Star-CCM+ генератора сеток. Толщина первого слоя призматического подслоя на модели гребного винта  $y_1^+$  составляет  $\sim 0.5$ , на корпусе модели судна – 80-100. Во всех расчетах используется модель турбулентности  $k-\omega$  SST [13] в комбинированном (высоко- и низкорейнольдсовом) варианте.

В проверочных расчетах, описанных далее в разделе «Проведение контрольных самоходных испытаний» дополнительно используется модель ламинарно-турбулентного перехода [14], модель Volume of Fluid (VOF) [15] для моделирования многофазных течений (свободная поверхность и кавитационная каверна), и модель кавитации Schnerr and Sauer [16].

Обоснование использованной модели течения, демонстрирующей высокую точность современных расчетных методов, представлено в работе [17], посвященной всесторонней валидации комплекса технологий численного моделирования на примере решения задач, связанных с процессом обтекания контейнеровоза KCS и его гребного винта потоком вязкой жидкости (определение буксировочного сопротивления и посадки судна, определение кривых действия гребного винта, определение номинального поля скоростей в диске гребного винта, численное моделирование самоходных испытаний, определение сопротивления судна и элементов его посадки на встречном волнении), выполненной на основе сравнения с результатами экспериментальных исследований KRISO.

Результаты валидации технологии численного моделирования кавитационного обтекания моделей судовых движителей, как на примерах международных тестовых объектов, так и на примерах решения практических задач в области проектирования гребных винтов представлены в работе [18].

В соответствии с условиями теста № 2.5 Токийского симпозиума 2015 года, численное моделирование самоходных испытаний модели KCS, оборудованного гребным винтом, выполняется при фиксированной скорости буксировки  $U=2.196$  м/с и фиксированной частоте вращения вала  $n=9,5$  об/сек. Посадка модели судна зафиксирована в положении на ровный киль при отсутствии всплытия. Определяются коэффициент сопротивления судна в самоходных испытаниях  $C_{TSp} = \frac{R_{TSp}}{\frac{1}{2}\rho U^2 S}$ , коэффициенты упора ( $K_T$ ) и момента ( $K_Q$ ) гребного винта, а также сила реакции буксировочной тележки  $Z$ , представляющая собой разницу между сопротивлением судна в условиях самоходных испытаний ( $R_{TSp}$ ) и упором гребного винта ( $T$ ). Подробно результаты валидации технологии численного моделирования самоходных испытаний представлены в работе [17].

### 3. Постановка задачи оптимизации модели гребного винта, работающего за корпусом судна

В качестве целевых функций выбраны увеличение КПД ( $\eta$ ) двигателя на заданном режиме хода судна и снижение пульсаций давления в пространстве около гребного винта. При оптимизации гребного винта модели КР505 были наложены ограничения для величин упора, момента и дискового отношения двигателя.

$$T_{opt} \geq T_{base} ,$$

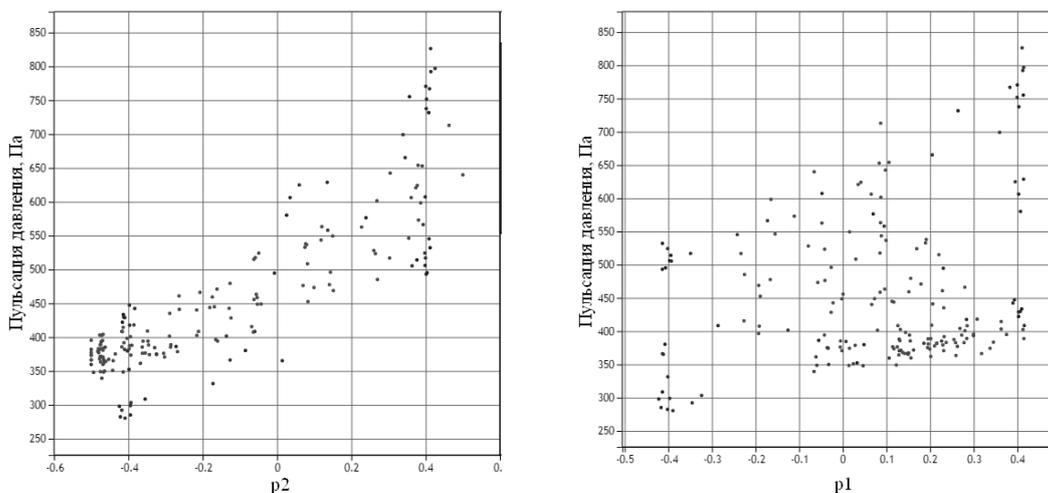
$$0.99 * Q_{base} \leq Q_{opt} \leq 1.01 * Q_{base} ,$$

$$A_E/A_{0_{opt}} \geq 0.9 * A_E/A_{0_{base}} ,$$

где  $T_{base}$  – значение упора гребного винта КР505 (Н),  $T_{opt}$  – значение упора гребного винта после оптимизации (Н),  $Q_{base}$  – значение момента гребного винта КР505 (Н·м),  $Q_{opt}$  – значение момента гребного винта после оптимизации (Н·м),  $A_E/A_{0_{base}}$  – значение дискового отношения гребного винта КР505,  $A_E/A_{0_{opt}}$  – значение дискового отношения оптимизированного гребного винта. Контроль величины дискового отношения необходим при изменении ширины лопасти гребного винта во избежание ухудшения кавитационных свойств модели двигателя. Хотя оптимизация должна проводиться при условии сохранения заданной мощности на валу (заданный момент при фиксированной частоте вращения), для момента задан допустимый диапазон. Это обусловлено возможным возникновением проблем при проведении оптимизации в случае задания ограничений в виде фиксированных величин. Диапазон допустимого изменения момента выбран исходя из допустимой погрешности в определении спецификационной скорости (с существенным запасом). В принципе, при проведении расчетов для реальной конструкции, возможно существенное уменьшение этого диапазона.

С целью сокращения параметров, изменяемых в ходе оптимизации, был проведен предварительный анализ влияния параметров модели гребного винта КР505 на выбранные целевые функции. Для этого процедура оптимизации с расширенным набором параметров проводилась в фиксированном поле скоростей, полученном при моделировании самоходных испытаний модели КС с работающим винтом КР505. Данная постановка задачи позволяет сократить требуемые вычислительные ресурсы и увеличить за счет этого количество модифицируемых параметров.

В результате проведенного предварительного анализа были отобраны группы параметров, влияющих на определенные в рамках данной работы критерии (Рисунок 3). Учитывалось как влияние на группу критериев, так и на каждый критерий по отдельности. Выбранные для подробных оптимизационных исследований параметры, критерии и ограничения представлены в Таблице 1.



**Рис. 3** Примеры выраженной (слева) и невыраженной (справа) зависимостей критерия от параметра оптимизационного цикла.

Для оптимизации используется расчетная сетка размерностью около 6,1 млн. ячеек, в т.ч. 3,8 млн. ячеек во вращающемся регионе, как наиболее экономичная по результатам исследования сеточной сходимости в задачах определения ГДХ движителей [19]. Подробно процесс влияния качества расчетной сетки и как следствие корректности результатов численного моделирования рассмотрен в работе [19]. Для сокращения размерности расчетной сетки при проведении оптимизационных исследований в расчетной схеме отсутствует учет свободной поверхности (используется дублированная модель судна). Также для ускорения расчетов в процессе оптимизации не использована модель ламинарно-турбулентного перехода.

**Таблица 1.** Условия задач нахождения оптимальных параметров модели гребного винта.

Номер оптимизационного поиска	1	2	3
Критерий оптимизационного поиска	Повышение $\eta$ на заданном режиме	Снижение пульсаций давления	Повышение $\eta$ на заданном режиме + снижение пульсаций давления
Набор управляемых параметров	Ширина лопасти (2), макс. толщина лопасти (2), саблевидность лопасти (1)	Саблевидность лопасти (1), ширина лопасти (2), шаг (2)	Саблевидность лопасти (1), ширина лопасти (2), шаг (2)
Ограничения	$T_{opt} \geq T_{base},$ $0.99 * Q_{base} \leq Q_{opt} \leq 1.01 * Q_{base},$ $A_E/A_{0opt} \geq 0.9 * A_E/A_{0base}$		

#### 4. Результаты оптимизации гребного винта.

Для проверки устойчивости оптимизационного алгоритма выполняются три запуска оптимизационного поиска №1 с различными начальными условиями. Запуск №1 выполняется с начальной формой гребного винта, соответствующей модели КР505. Запуски №2 и №3 выполняются с типичными формами гребных винтов транспортных судов (см. Рисунок 5).

На Рисунке 5 также приведен общий вид полученных форм гребного винта. Контролируемые интегральные характеристики гребного винта представлены в Таблице 2. Распределение локальных параметров, изменённых в ходе оптимизации, представлены на Рисунке 4. Необходимость выполнения условий прочности гребного винта допускает в процессе оптимизации лишь незначительные изменения толщины лопасти, поэтому распределение толщины, а также прочих, зафиксированных в процессе оптимизации параметров таких, как профилировка лопасти, не приводятся.

**Таблица 2.** Результаты оптимизации формы гребного винта КР505. Критерий – повышение КПД

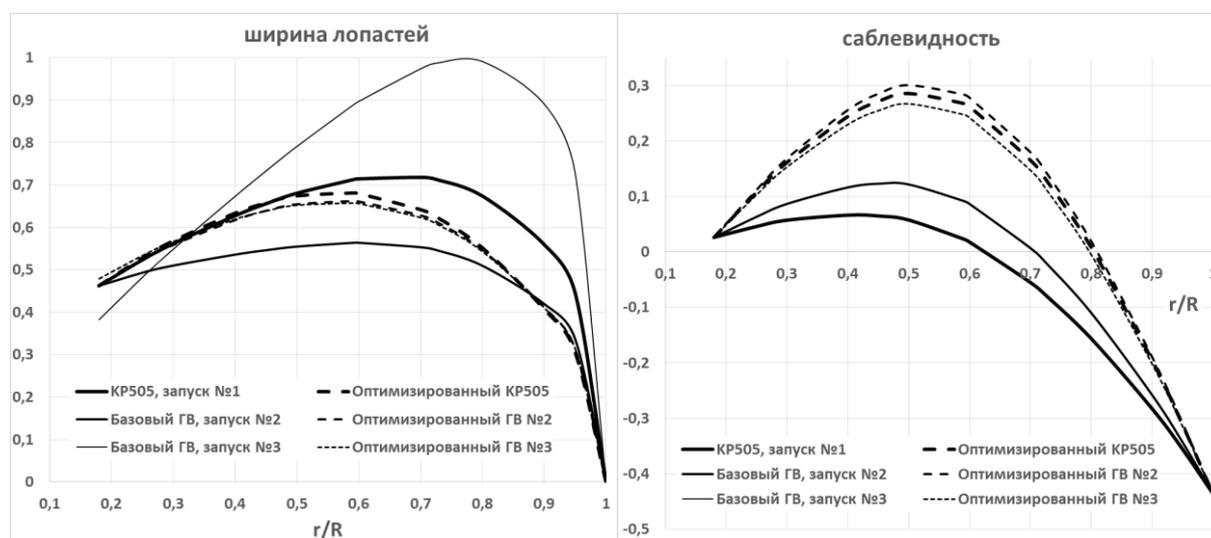
Характеристика	Начальные значения КР505	Результат оптимизационного поиска №1		
		Запуск №1	Запуск №2	Запуск №3
$A_e/A_0$	0,80	0,733 (-8,4%)	0,720 (-10,0%)	0,721 (-9,88%)
$T, H$	61,96	64,04 (+3,34%)	64,14 (+3,50%)	64,15 (+3,52%)
$Q, H^*m$	2,794	2,795 (+0,03%)	2,794 (-0,02%)	2,796 (+0,05%)
$\eta$	0,815	0,842 (+3,31%)	0,844 (+3,53%)	0,843 (+3,47%)

Как видно из Рисунков 4 и 5, независимо от начального приближения оптимизационный алгоритм находит практически идентичное решение, соответствующее граничным условиям, критерию поиска и ограничениям задачи.

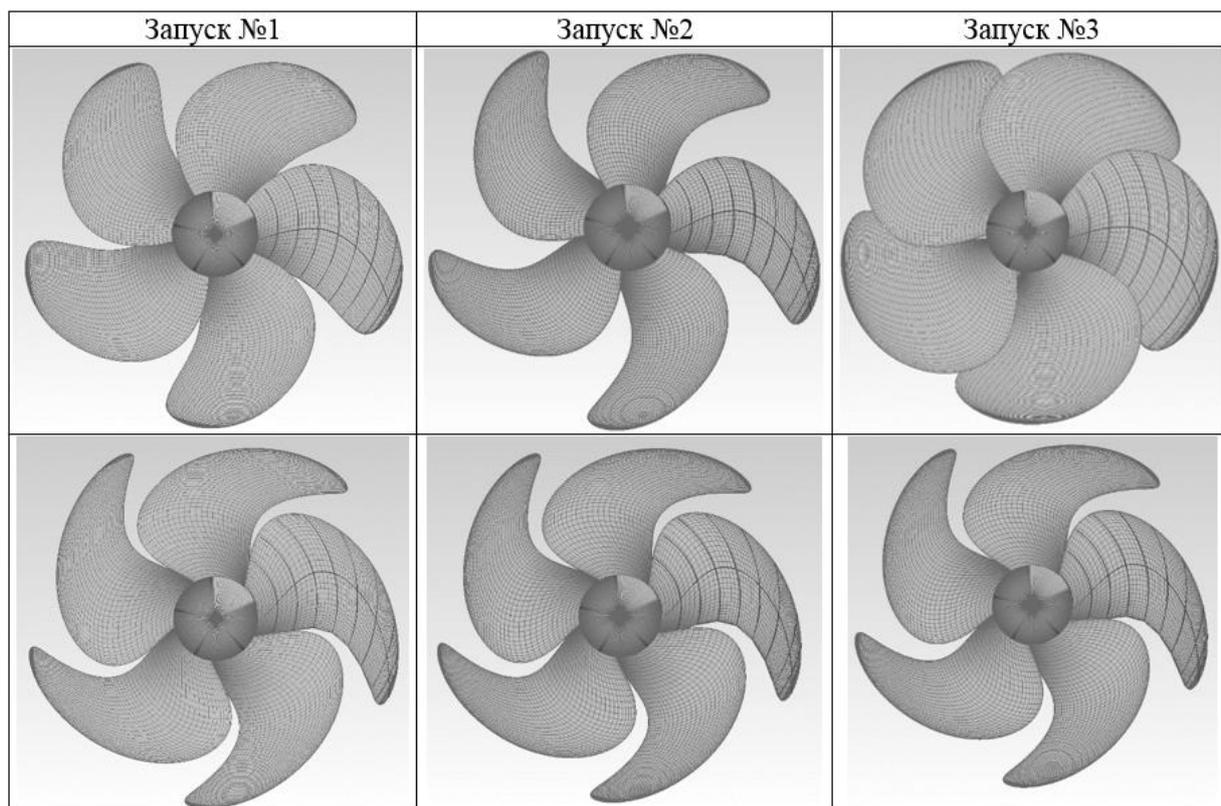
Каждый процент КПД двигателя соответствует в среднем 1% мощности энергетической установки или 0.3% скорости хода водоизмещающего судна. Таким образом полученный выигрыш по КПД и, соответственно скорости хода, сравним с условиями применения штрафных санкций по скорости хода (0.3 узла) на сдаточных испытаниях транспортных водоизмещающих судов.

Следующей частью исследования является проверка гребного винта проекта КР505 и гребного винта, полученного в ходе оптимизации (Запуск №1), на соответствие требованиям Российского морского регистра судоходства (РМРС) в части конструкции и размеров двигателей (раздел 6, глава 6.2) [20]. Рассматриваются гребные винты натурального размера с масштабом, предложенным проектантом контейнеровоза KCS [5]. В Таблице 3 приведены значения требуемой по РМРС и фактической толщины лопасти гребных винтов на расчетном радиусе  $r/R=0,6$ . Толщина и ширина лопасти оптимизированных гребных винтов на расчетном радиусе  $r/R=0,2$  не изменялась по сравнению с исходным гребным винтом проекта КР505.

Как видно из Таблицы 3, оптимизированный гребной винт имеет меньший запас прочности, чем исходный, удовлетворяя тем не менее требованиям РМРС в этой части.



**Рис. 4** Сравнение распределений ширины (слева) и саблевидности (справа) лопастей гребных винтов, использованных в качестве начальных условий (сплошные линии) и оптимизированных для работы в поле контейнеровоза KCS (пунктир).



**Рис. 5** Формы гребных винтов, использованных в качестве начальных условий (вверху), и формы гребных винтов, оптимизированных для работы в поле контейнеровоза KCS (внизу).

**Таблица 3.** Сравнение требуемой и фактической толщины лопасти гребного винта на расчетном радиусе  $r/R=0,6$ .

Характеристика	Гребной винт КР505	Результат оптимизации
Требуемая толщина лопасти на расчетном радиусе, мм	127,6	134,4
Фактическая толщина лопасти на расчетном радиусе, мм	153,8	143,3
Запас толщины	20,6%	6,6%

Аналогично выполняются оптимизационные поиски №2 и №3 (см. Таблицу 1). Для оценки уровня пульсаций давления в расчетную область дополнительно помещено облако контрольных точек, в которых в ходе оптимизационного цикла минимизируются максимальные пульсации давления.

Результаты оптимизационных поисков №1-3 характеризуются близкими формами полученных движителей, однако анализ распределений параметров показывает, что снижение пульсаций давления получается в первую очередь из-за сброса шага на конце лопасти, а повышение КПД за счет совместного изменения ширины лопасти и её саблевидности.

## 5. Контрольное моделирование самоходных испытаний

Для комплексной оценки эффективности оптимизированных гребных винтов с их использованием проводится контрольное моделирование самоходных испытаний на основе полной модели, описанной в разделе «Постановка задачи определения гидродинамических характеристик судна и движителя. Обоснование выбранных моделей течения», т.е. с учетом свободной поверхности и ламинарно-турбулентного перехода на гребном винте. Размерность расчетной сетки составляет 7.65 млн. ячеек, из них 3.8 млн. во вращающемся регионе. Дополнительное сгущение расчетной сетки выполнено в районе свободной поверхности. Результаты численного моделирования самоходных испытаний представлены в Таблице 4.

Как видно из Таблицы 5, по результатам численного моделирования самоходных испытаний с использованием подробной расчетной сетки и более сложной расчетной схемы все оптимизированные гребные винты обладают более высоким КПД, чем исходный. Превышение момента вращения над исходным составляет не более 1%. Для оптимизационного поиска №2 снижение потребного момента вращения оказалось больше на 1%, чем предполагалось. Этот факт объясняется нелинейным влиянием модели ламинарно-турбулентного перехода (не учитываемой непосредственно в ходе оптимизации) на изменение трения лопастей гребного винта.

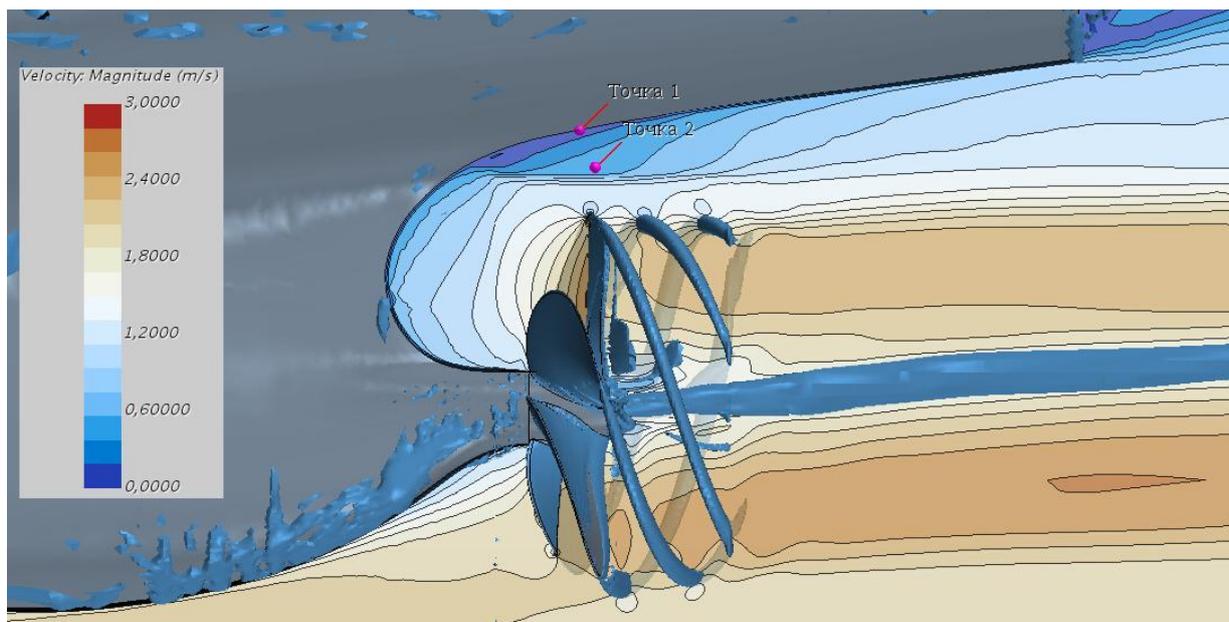
В рамках данной работы оценивалась только работоспособность разрабатываемой методики оптимизации применительно к гребному винту. Задача оптимизации по коэффициентам взаимодействия гребного винта и корпуса судна (коэффициенты попутного потока и засасывания) не ставилась. Однако данные Таблицы 4 показывают, что, хотя использование оптимизированных винтов приводит к некоторому увеличению сопротивления корпуса ( $R$ ) и момента сопротивления вращению гребного винта, в целом оптимизированные винты обеспечивают выигрыш по потребляемой мощности. Увеличение  $R$ , трактуемое как увеличение засасывания, обусловлено увеличением упора гребного винта. Коэффициент засасывания при этом практически не изменяется, как и коэффициент влияния корпуса  $\eta_n$ . Что соответствует многократно подтвержденной практикой проведения самоходных испытаний с «магазинными» гребными винтами.

Для косвенной оценки виброакустических характеристик гребного винта в расчетной области установлены две точки мониторинга давления. Первая расположена на поверхности корпуса в диаметральной плоскости над гребным винтом и имеет координаты (0.13 м, 0.0 м, -0.0307 м). Вторая находится в пространстве между гребным винтом и корпусом судна и имеет координаты (0.12 м, 0.0 м, -0.055 м). Положение точек в пространстве показаны на Рисунке 6 (Точка 1 и Точка 2).

В Таблице 6 приведены значения средних уровней давления и его относительных пульсаций (амплитуда пульсаций) в точках мониторинга для рассмотренных моделей гребных винтов. Осреднение производится за 10 оборотов гребного винта. Как видно из Рисунка 7 и Таблицы 5, все три варианта оптимизированных гребных винтов вызывают меньшие пульсации давления в точках мониторинга за счет увеличенной саблевидности. Средние значения давления в точках мониторинга незначительно выросли за счет увеличившейся нагрузки гребного винта.

**Таблица 4.** Результаты контрольных самоходных испытаний.

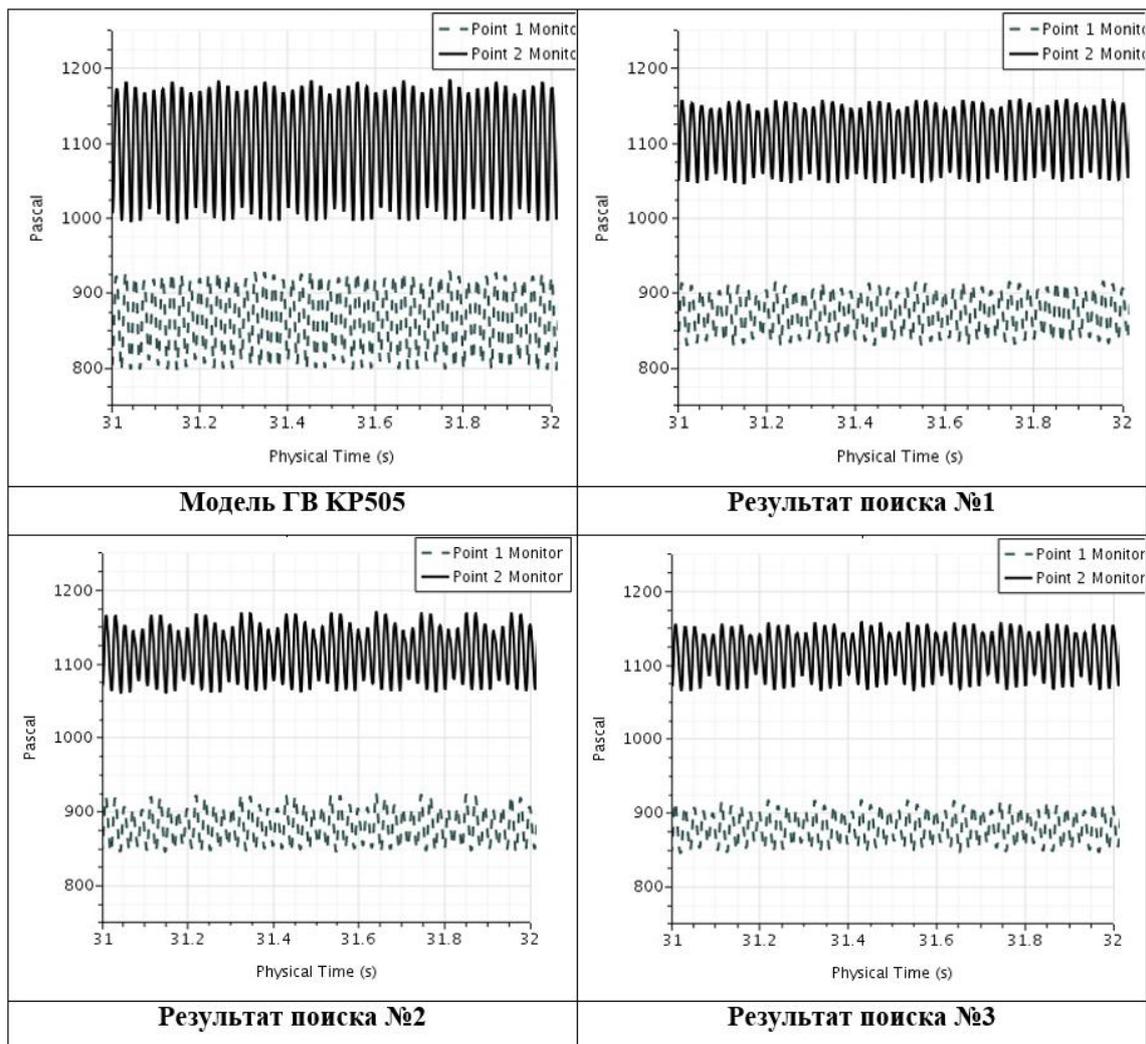
Характеристика	Гребной винт КР505	Результат оптимизации		
		Поиск №1	Поиск №2	Поиск №3
$R, Н$	-90,71	-91,26	-91,46	-91,66
$T, Н$	62,52	64,24	61,48	63,95
$Q, Н*м$	2,698	2,717	2,636	2,712
$Z, Н$	-28,19	-27,04	-29,98	-27,71
$C_T$	$3,991*10^{-3}$	$4,015*10^{-3}$	$4,024*10^{-3}$	$4,033*10^{-3}$
$K_T$	0,17750	0,18233 (+2,72%)	0,17455 (-1,66%)	0,18156 (+2,29%)
$K_Q$	0,03064	0,03086 (+0,72%)	0,0299 (-2,29%)	0,0308 (+0,52%)
$\eta$	70,6 %	72,0 % (+1,4%)	71,1 % (+0,46%)	71,9 % (+1,2%)



**Рис. 6** Визуализация вихревых структур, сходящих с движителя ( $Q=1000$ ) и поле скорости в диаметральной плоскости судна.

При этом минимальные пульсации в точках мониторинга достигнуты при использовании гребного винта, оптимизированного по двум критериям (поиск №3), а не по критерию снижения пульсаций давления (поиск №2), как предполагалось. Это объясняется тем фактом, что абсолютные значения пульсаций в точках мониторинга при проведении контрольных самоходных испытаний существенно (в 3-4 раза) меньше, чем абсолютные значения пульсаций в целевой функции, построенной по облаку точек. Таким образом, требуется дополнительная проработка критериев в части корректной оценки виброакустических характеристик гребного винта в процессе оптимизации. Одним из возможных подходов является минимизация объемов пониженного давления вблизи гребного винта.

В целях проверки возможного ухудшения кавитационных характеристик гребного винта, полученного в ходе оптимизационной процедуры, в расчетную схему включена модель кавитации [16]. Расчеты выполнялись для натурального числа кавитации  $\sigma_n=1.834$  (масштаб 1:31.6). Такой подход соответствует современной практике проектирования гребных винтов, когда кавитационные характеристики движителя проверяются в кавитационных трубах непосредственно с использованием модели корпуса [21].

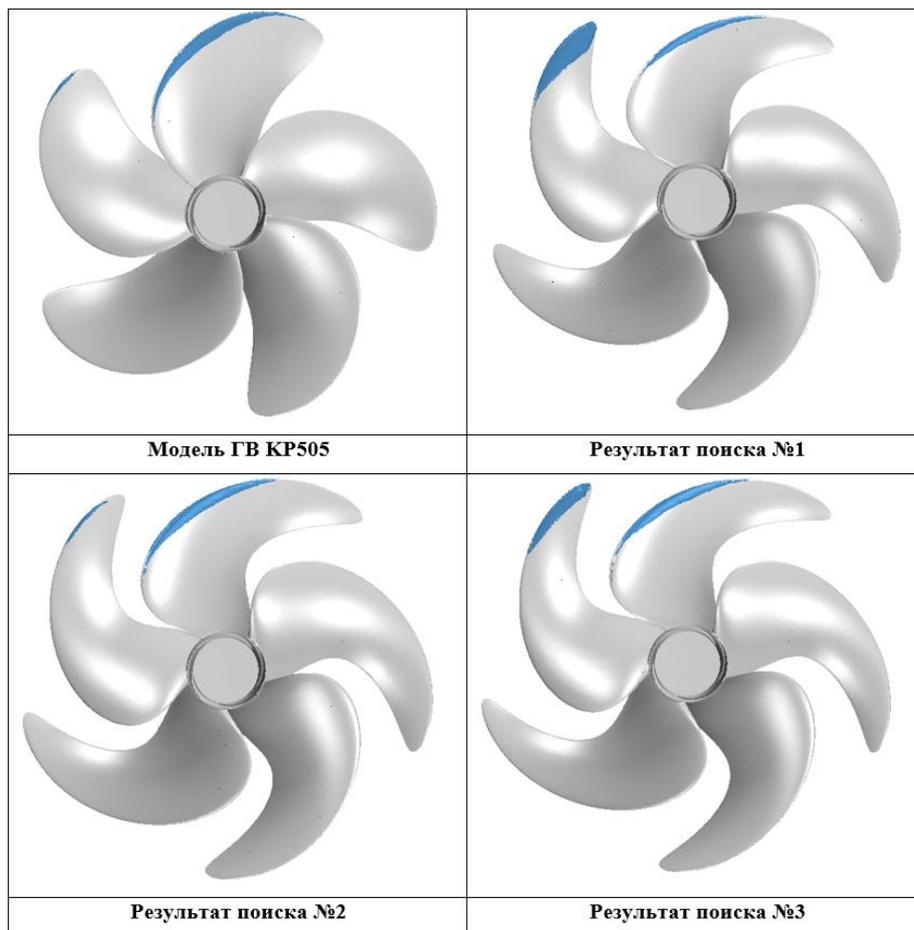


**Рис.7** Зависимость от времени пульсаций давления в точках мониторинга. Гребной винт КР505 и гребные винты, полученные при различных условиях оптимизационного поиска (Таблица 1).

**Таблица 5.** Средние уровни давления и его относительные пульсации.

Характеристика	Гребной винт КР505	Результат оптимизационного поиска		
		Поиск №1	Поиск №2	Поиск №3
Среднее давление в точке мониторинга №1, Па	866	873	880	880
Средний уровень пульсаций давления в точке мониторинга №1	15,5 %	9,96 %	9,02	8,2%
Среднее давление в точке мониторинга №2, Па	1099	1108	1115	1115
Средний уровень пульсаций давления в точке мониторинга №2	17,34 %	10,24 %	9,98	8,45%

Вид кавитационных каверн для исходного и оптимизированных гребных винтов представлен на Рисунке 8. Средний объем кавитационных каверн (осреднение за несколько оборотов) представлен в Таблице 6. Проверка запасов на вторую стадию кавитации с проведением соответствующих расчетов не выполнялась, однако можно предположить, что они если и уменьшились, то незначительно в силу относительно малых абсолютных величин объемов кавитационных каверн.



**Рис. 8** Кавитационные каверны на модели гребного винта. Гребной винт КР505 и гребные винты, полученные при различных условиях оптимизационного поиска.

**Таблица 6.** Осреднённый объем кавитационных каверн на лопастях гребных винтов.

Гребной винт КР505	Результат оптимизационного поиска		
	Поиск №1	Поиск №2	Поиск №3
0,42 см <sup>3</sup>	0,82 см <sup>3</sup>	0,46 см <sup>3</sup>	0,69 см <sup>3</sup>

## 6. Вычислительные ресурсы

Для выполнения численного моделирования в процессе оптимизации и выполнения контрольных расчётов использован суперкомпьютер ФГУП «Крыловский государственный научный центр». Использованный сегмент суперкомпьютера состоит из сорока двухпроцессорных блейд-серверов на основе 12-ти ядерного процессора Intel Xeon E5-2697 v2 с тактовой частотой 2.7 ГГц объединенных высокоскоростной сетью с низкими задержками Mellanox QDR Infiniband с топологией «толстое дерево». Частота оперативной памяти 1600 МГц.

**Таблица 7.** Вычислительные ресурсы, затраченные на оптимизацию модели гребного винта КР505

	1 критерий	2 критерия
Время, затраченное на 1 запуск	~45 минут	
Количество запусков	188	810
Общее время оптимизации формы ГВ	~6 суток	~25 суток
Вычислительные ресурсы	480 ядер Пиковая производительность 11,2 ТФлопс	

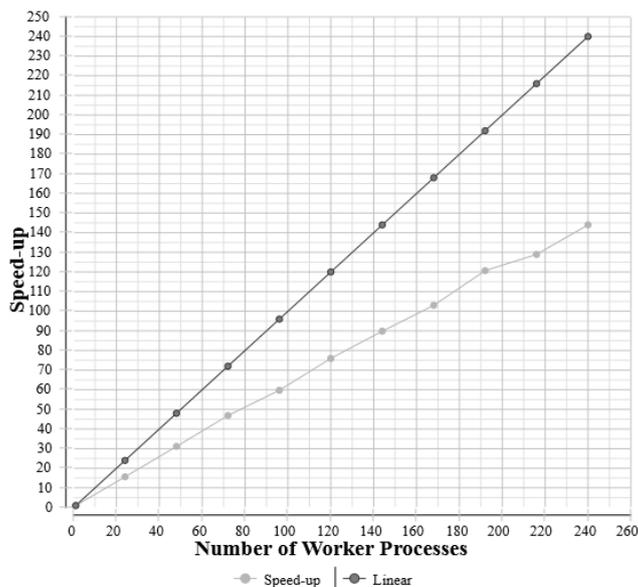


Рис. 9 Эффективность распараллеливания задачи об обтекании корпуса судна, решаемой в пакете Star-CCM+ 13.02.013 на использованном сегменте суперкомпьютера.

## Заключение

В ходе выполнения оптимизационных исследований, направленных на повышение эффективности гребного винта контейнеровоза KCS, был получен ряд новых форм движителя. Полученные решения характеризуются пониженным уровнем пульсаций давления в кормовой оконечности судна, а также повышенным упором и КПД, что достигается за счет возможного незначительного снижения запасов на кавитацию и запасов прочности. При этом выполняется ограничение на располагаемый момент вращения и толщину лопастей по правилам РМРС.

В целом проделанная работа демонстрирует работоспособность разрабатываемой в Крыловском государственном научном центре технологии оптимизации движителя с учетом влияния корпуса судна. Заявленные характеристики технологии подтверждены на подробных расчетных моделях, учитывающих наиболее принципиальные факторы. Дополнительная валидация технологии на основе сопоставления с результатами физического моделирования будет выполнена в рамках текущей опытно-конструкторской работе по Государственной программе «Развитие судостроения и техники для освоения шельфовых месторождений» и опубликована отдельно.

В ближайших планах по развитию технологии находится включение проверочных процедур на риски возникновения кавитации [22] и оценки прочности, как по расчетной методике Российского морского регистра судоходства, так и с использованием конечно-элементного метода, внутрь оптимизационного цикла.

Ключевым моментом дальнейшего развития технологии оптимизации является переход к ее выполнению для натуральных условий. Именно переход к использованию RANS позволяет это сделать наиболее корректно в отличие от идеальной жидкости, используемой другими исследователями. Однако такое принципиальное изменение подхода к выполнению оптимизационной процедуры возможно только после ее тщательной валидации для условий модельного эксперимента.

## Литература

1. Левковский Ю.Л. Структура кавитационных течений, Л.: Судостроение, 1978. – 224 с.
2. Proceedings of Fifth International Symposium on Marine Propulsors (SMP), Volume 2, Session 5A Design/Optimization, Espoo, Finland, 2017. P. 258-290.

3. Лаврищева Л.С., Новоселов В.Н. Оптимизация формы модели гребного винта в однородном потоке. Труды Крыловского государственного научного центра. 2018; специальный выпуск 1: 75–83.
4. Лаврищева Л.С. Численное моделирование обтекания объектов морской техники и разработка технологии оптимизации формы гребного винта // Диссертация на соискание ученой степени кандидата физико-математических наук, Санкт-Петербург, - 2018. – 246 с.
5. Van, S.H., Kim, W.J., Yim, G.T., Kim, D.H., Lee, C.J. Experimental Investigation of the Flow Characteristics Around Practical Hull Forms. Proceedings 3rd Osaka Colloquium on Advanced CFD Applications to Ship Flow and Hull Form Design, Osaka, Japan, 1998
6. Kim, W.J., Van, D.H., Kim, D.H. Measurement of flows around modern commercial ship models. Exp. in Fluids, 2001; 31: 567-578.
7. Hino, T. Proceedings of CFD Workshop Tokyo 2005, NMRI report 2005
8. Официальный сайт компании ООО «ДАТАДВАНС»[Электронный ресурс]. URL: <https://www.datadvance.net/ru/>
9. Forrester A., Recent advances in surrogate-based optimization / Alexander Forrester, Andy Keane // Progress in aerospace sciences.–Elsevier, Southhampton, UK.–2009. P. 1 –77.
10. Свидетельство № 2019663281 Российская Федерация. Программа для ЭВМ «Balde Plus»: свидетельство об официальной регистрации программы для ЭВМ / Н. В. Новоселов; заявитель и правообладатель ФГУП «Крыловский государственный научный центр» - № 2019662332; заявл. 07.10.2019; зарегистр. 15.10.2019. - 1 с.
11. Evert-Jan Foeth, Propeller optimization using an unsteady Boundary-Element Method. Fourth International Symposium on Marine Propulsors (SMP), 2015.
12. [http://www.nmri.go.jp/institutes/fluid\\_performance\\_evaluation/cfd\\_rd/cfdws15/file/Geometry\\_IGES\\_files/kcs/KP505.zip](http://www.nmri.go.jp/institutes/fluid_performance_evaluation/cfd_rd/cfdws15/file/Geometry_IGES_files/kcs/KP505.zip)
13. Menter F.R. Two equation eddy-viscosity turbulence models for engineering applications // AIAA Journal. 1994, Vol.32. pp. 1598-1605.
14. Menter F.R., Langtry R.B., Likki S.R., Suzen Y.B., Huang P.G., Völker S.A. Correlation-based transition model using local variables. Part 1: Model Formulation // ASME J. Turbomachinery. 2006; 128(3): 413–22.
15. Hirt C.W., Nichols B.D. Volume of fluid (VOF) method for the dynamics of free boundaries // Journal of Computational Physics. 1981; 39(1): 201–25.
16. Sauer J. Instationaer kavitierende Stroemungen - Ein neues Modell, basierend auf Front Capturing VOF und Blasendynamik. Dissertation. Universitaet Karlsruhe, 2000.
17. Таранов А.Е., Определение локальных и интегральных гидродинамических характеристик контейнеровоза в цифровом бассейне. Труды Крыловского государственного научного центра. 2019; 3(389): 73-82.
18. Bagaev D., Yegorov S., Lobachev M., Rudnichenko A. and Taranov A. Numerical Simulation of Cavitating Flows in Shipbuilding, International Scientific Conference on Mechanics - Eighth Polyakhov's Reading, AIP Conference Proceedings 1959, 050003 (2018); doi: 10.1063/1.5034631
19. Таранов А.Е., Сеточная сходимость в расчетах обтекания гребного винта ледокола, Труды Крыловского государственного научного центра, 90(374), 2015, стр. 55-62
20. Правила классификации и постройки морских судов, Часть VII, Механические установки, Российский Морской Регистр Судоходства, Санкт-Петербург, 2018
21. Cavitation tunnel SSPA [Электронный ресурс], <https://www.sspa.se/our-facilities-and-tools/cavitation-tunnel>

22. Лобачев М.П., Рудниченко А.А. Расчетное определение начала кавитации рабочего колеса водометного движителя насосного типа. Труды Крыловского государственного научного центра. 2019; 3(389): 63–72.

# Моделирование влияния системы мониторинга производительности на выполнение коллективных MPI операций<sup>1\*</sup>

А.А. Худолеева, К.С. Стефанов

МГУ имени М.В. Ломоносова

Изучение параллельных программ с помощью средств мониторинга производительности — распространенная практика. Агент системы мониторинга для сбора данных о работе приложения периодически активируется во время счета этого приложения, внося помехи и занимая ресурсы. Однако вопрос об уровне влияния этих помех является слабо изученным, разработчики систем мониторинга зачастую не проводят исследования в этом направлении. В данной статье рассматриваются подходы к изучению влияния системы мониторинга производительности суперкомпьютера на пользовательские приложения. В качестве инструмента для измерения влияния агента системы мониторинга предлагается использовать коллективные MPI операции. Время выполнения коллективных MPI операций изучается в присутствии программного средства, моделирующего работу агента системы мониторинга. Оценивается уровень шума, который каждая из рассматриваемых коллективных операций в выбранной конфигурации запуска способна зафиксировать. В работе приводятся данные запусков инструмента с коллективными MPI операциями All-to-All, All-Reduce, Barrier. Найдено, что хорошей стабильностью и чувствительностью обладают операции All-to-All и Barrier.

*Ключевые слова:* суперкомпьютер, мониторинг производительности, шум системы мониторинга, замедление параллельных задач, моделирование влияния системы мониторинга.

## 1. Введение

Современные суперкомпьютеры — это очень сложные, большие вычислительные комплексы, состоящие из множества компонентов. Рост возможностей системы, под которую пишется приложение, затрудняет его разработку и усложняет структуру этого приложения. Это ведет к повышению сложности оптимизации приложения под целевую платформу. Достижение максимальной производительности и наиболее полное использование предоставленных ресурсов — важная задача. Невозможность эффективно использовать средства суперкомпьютера ведет к замедлению работы программ и, следовательно, к задержке проведения научного исследования. Для решения данной проблемы оптимизации приложения можно использовать специальный продукт — средство мониторинга производительности.

При проектировании средства мониторинга разработчики преследуют цель создания масштабируемого, эффективного инструмента, который будет предоставлять для анализа необходимую информацию об используемых пользовательским процессом ресурсах. Данные могут характеризовать, например, взаимодействие приложения с различными частями системы: с процессором, сетью, памятью.

Агент системы мониторинга запускается совместно с исследуемым приложением на тех же вычислительных узлах и с определенным периодом выполняет необходимые действия: собирает данные и отправляет их для дальнейшей обработки. Так как средство мониторинга разделяет ресурсы с приложением, оно вносит помехи в работу исследуемого приложения и негативно влияет на производительность. К данному моменту разработано достаточное количество различных систем мониторинга. Однако авторы этих систем часто утверждают, что влияние

---

\* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00940.

Работа выполнена с использованием оборудования Центра коллективного пользования сверхвысоко-производительными вычислительными ресурсами МГУ имени М.В. Ломоносова

средств мониторинга незначительно, не приводя описания методов, позволяющих это влияние измерить.

Вопросу обнаружения влияния системы мониторинга производительности суперкомпьютера с помощью использования коллективных MPI операций и установления уровня этого влияния указанным методом посвящена эта работа.

## 2. Обзор области

За последние 20 лет было уделено много внимания созданию различных средств мониторинга производительности: Supermon [1], NWPerf [2], HPCToolkit [3], Performance Co-Pilot [4], LIKWID [5,6], LDMS [7]. Авторы статей об инструментах HPCToolkit, Performance Co-Pilot, LIKWID говорят, что эффект, оказываемый этими средствами мониторинга на пользовательские приложения, является незначительным, однако не приводят результатов исследований, связанных с этим вопросом.

Агентом системы мониторинга NWPerf [2] является модуль ядра Linux. Обычный интервал сбора данных — одна минута. Для этого инструмента было проведено исследование его влияния на коллективные операции All-to-All и All-Reduce. На 128 узлах, на 256 вычислительных ядрах запускался цикл из 10,000 коллективных операций с и без средства мониторинга. При частоте сбора данных в десять раз превышающей стандартный режим (1 раз в 6 секунд) время выполнения операции All-to-All увеличилось на 27%, а операции All-Reduce — на 9.46%. Но при стандартной частоте прочие помехи в системе перекрывали шум от NWPerf, что свидетельствует о приемлемости частоты работы агента раз в минуту.

Достаточно подробно было исследовано влияние средства мониторинга LDMS [7] на работу различных компонент высокопроизводительной системы. Средой проведения экспериментов выступали две большие вычислительные системы. Тестирование с использованием бенчмарков показало, что влияние инструмента LDMS имеет незначительный эффект на производительность MPI коммуникаций.

Разница в полученных результатах в вопросе влияния системы мониторинга на MPI коммуникации значительная. Она может быть вызвана аппаратными и сетевыми отличиями суперкомпьютеров, на которых проводились исследования. Вопрос влияния помех, вносимыми системами мониторинга, на производительность параллельных программ остается открытым.

Хорошо исследованной является область обнаружения уровня влияния шума операционной системы на работу программ. Методы, применяемые в этой области, потенциально можно распространить на изучение шума системы мониторинга. Однако эти методы изучают то, что происходит на отдельных вычислительных узлах, в то время как многие свойства параллельных программ проявляются только при изучении ее как целого.

В работе [9] при использовании большого количества процессов удалось установить, что несинхронизированный шум, возникающий с периодической частотой, является причиной значительного замедления коммуникаций Barrier и All-Reduce. В исследовании [8], посвященном оптимизации приложения SAGE на суперкомпьютере ASCI Q, было показано, что источником замедления работы приложения была производительность операции All-Reduce. Размещая особым образом задачу на узлах суперкомпьютера, исследователи исключили появление частого синхронного шума, возникающего на некоторых узлах. Авторы исследования [8] сделали вывод о том, что на некоторые хорошо синхронизированные приложения небольшой, но частый, синхронный шум оказывает сильное влияние.

## 3. Методология исследования

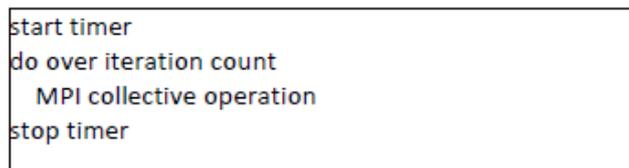
Одним из самых популярных средств при программировании приложений для вычислительных кластеров является MPI — интерфейс, поддерживающий работу параллельных процессов с помощью передачи сообщений между ними. Из функционала, который доступен при использовании MPI, широко используются коллективные операции. В исследовании [10] проводился анализ 110 параллельных приложений с открытым исходным кодом. Минимальный функционал, который используют все рассмотренные в работе приложения — вызов коллективных операций.

Приложение может выполняться несимметрично, то есть в зависимости от ранга, процессам назначаются роли, реализованные разными фрагментами кода. На большом вычислительном комплексе нельзя гарантировать бесперебойность работы каждого узла, симметрию выполнения приложения на каждом ядре. Поэтому даже в приложениях, где процессы выполняют один и тот же фрагмент кода, стартовав синхронно, время выполнения последовательной части программы будет разным для разных процессов. Так как при запуске коллективной операции все процессы должны одновременно выполнить пересылку сообщений или синхронизацию, то часть процессов будет простаивать, ожидая готовности к выполнению операции «опаздывающие» процессы. Поэтому время работы параллельного приложения измеряется по самому медленному из процессов.

В силу тех фактов, что использование коллективных MPI операций является распространенной практикой и они могут стать весомым источником замедления в работе программы, важным является вопрос рассмотрения влияния агента системы мониторинга на выполнение коллективных операций.

Задачей данной работы является исследование влияния системы мониторинга производительности на коллективные операции при использовании небольшого числа узлов суперкомпьютера. Возможность использования коллективных MPI операций, как средство для обнаружения шума, создаваемого системой мониторинга, и его уровня, исследуется ниже.

### 3.1 Программное средство, детектирующее шум



**Рис. 1.** Схема реализации инструмента для обнаружения шума

На рисунке 1 представлена схема реализации программного средства, с помощью которого предлагается исследовать влияние шума — детектор шума. Измеряемой величиной является время выполнения цикла из *count* числа коллективных MPI операций.

Рассматриваемыми коллективными MPI операциями в рамках данной работы являются: Barrier, All-Reduce, All-to-All.

Для разработанного программного средства существует несколько свободных параметров, выбрав которые, можно определить наилучшую конфигурацию эксперимента. Конфигурация инструмента считается подходящей, если при запуске в стандартных условиях, в отсутствие искусственно внедренного шума, эксперимент является воспроизводимым, то есть от запуска к запуску время выполнения коллективных операций изменяется в пределах допустимой погрешности. Также важной является способность инструмента фиксировать присутствие в системе искусственного шума, чем точнее чувствительность, тем лучше.

Свободные параметры включают в себя:

- выбор коллективной операции
- количество узлов, на которых запущено программное средство, количество процессов
- количество коллективных операций
- для операций All-Reduce и All-toAll длина передаваемого сообщения

### 3.2 Программное средство, моделирующее шум агента системы мониторинга

На рисунке 2 представлен фрагмент кода программы — источника шума, которая моделирует влияние агента системы мониторинга производительности на работу пользовательского приложения.

В основе работы этого средства лежит цикл, в котором вычисляется сумма (значение суммы хранится в переменной *sum*) массива *\*a* объемом 250 000 слов (значение константы *SIZE*) типа long long. В *WORK\_TIME* хранится доля секунды, которую работает нагрузка — суммиро-

вание массива. В параметр `tw` процедуры `nanosleep()` записана доля секунды, в течение которой сумма массива не вычисляется. Это значение дополняет `WORK_TIME` до 1 секунды. То есть  $100 \times \text{WORK\_TIME}$  — процент времени от `BREAK_TIME`, в течение которого моделируется присутствие шума. В дальнейшем под уровнем шума в процентах мы будем понимать значение  $100 \times \text{WORK\_TIME}$ .

Во фрагменте кода, реализующем генерацию нагрузки, отсутствуют операции синхронизации. С большой вероятностью выполнение такой программы будет асинхронным, что приведет к усилению влияния искусственного шума, как было отмечено при изучении шума операционной системы [9].

```
start_time = MPI_Wtime();
do
{
    t_1 = MPI_Wtime();
    do
    {
        for (int i = 0; i < SIZE; ++i)
        {
            sum += a[i];
        }
        t_2 = MPI_Wtime();
    }
    while ((t_2 - t_1) < WORK_TIME ;
    nanosleep(&tw, &tc);
    t_2 = MPI_Wtime();
}
while ((t_2 - start_time) < BREAK_TIME);
```

Рис. 1. Фрагмент реализации программного средства, моделирующего влияние агента системы мониторинга

### 3.3 Условия проведения экспериментов

Исследование разработанных программных средств было выполнено на суперкомпьютере Ломоносов-2 [11]. Запуски проводились на тестовом разделе суперкомпьютера. Так удалось избежать ожидания в загруженной очереди основного раздела. На каждом узле установлен процессор Intel Haswell-EP E5-2697v3, имеющий 14 ядер. Лимит времени выполнения заданий на тестовом разделе составляет 15 минут. Этим фактом обусловлен выбор некоторых из параметров тестируемых программных средств.

При запуске на суперкомпьютере использовалась библиотека OpenMPI [12].

### 3.4 Описание проведенных экспериментов

Исследование детектора шума проходило в два этапа.

Сначала, необходимо было оценить стабильность работы разных вариантов детектора шума. В случае существенного разброса времени выполнения коллективных операций на системе Ломоносов-2 при тестировании, нужно отказаться от такой конфигурации, так как она не подходит для многократного использования с целью измерения влияния шума. Причиной этого является отсутствие возможности получить стабильное значение времени работы операции без шума и, следовательно, отличить его от времени работы в присутствии сторонней нагрузки.

Далее выбранные варианты детектора шума запускались совместно с источником шума. Полученная разница во времени выполнения коллективных операций в этих двух случаях сравнивалась с помощью статистических критериев, описанных в работе [13].

В силу большого разнообразия возможностей проведения запусков была выбрана следующая конфигурация: программные средства запускались на 4 узлах суперкомпьютера Ломоносов-2, по 2 процесса на ядро (один процесс на виртуальное ядро, Hyper Threading включен). Для программы с операцией All-to-All также проводились тесты на 8 узлах, 2 процесса на ядро.

Модель агента системы мониторинга запускалась совместно с программным средством на тех же узлах, но на одном процессе одного ядра.

### 3.5 Результаты измерений

На графиках ниже представлены результаты тестов. Синим отмечены результаты «чистых» запусков, красным с шумом 1% от всего времени работы инструмента.

В таблицах 1, 2 представлены результаты измерений для каждого теста.

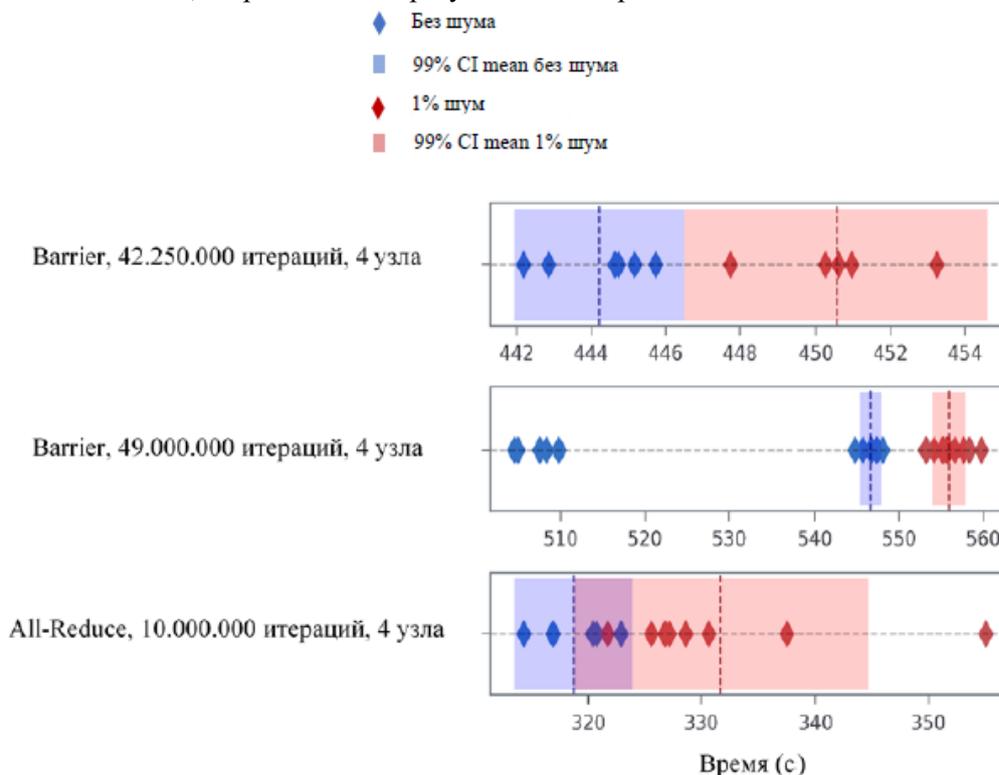


Рис. 3. Распределение времени для операций Barrier, All-Reduce

Таблица 1. Результаты запусков для I: Barrier, 42.250.000 итераций, 4 узла; II: Barrier, 49.000.000 итераций, 4 узла; III: All-Reduce, 10.000.000 итераций, 2KB, 4 узла.

		I	II	III
Чистые Запуски	Среднее время, с	444,22	546,66	318,67
	Стандартное Отклонение, с	1,39	1,03	3,19
	Количество Запусков	6	8	6
Запуски с шумом 1%	Среднее время, с	450,56	555,91	331,66
	Стандартное Отклонение, с	1,96	2,09	4,95
	Количество Запусков	5	11	8

Для 49.000.000 итераций операции Barrier, 4 узла на рисунке 3 видно, что показания программы сгруппированы около двух значений — 509 и 546 секунд. При построении таблицы 1 и доверительного интервала использовалась вторая группа. Наличие двух групп может быть объяснено тем, что задачи ставились в разные дни. Тем не менее, все значения времени для шума 1% превышают доверительный интервал времени без шума. Детектор с 42.250.000 итераций

операции Barrier будет пригоден для использования, если, увеличив число измерений, получится сузить доверительный интервал.

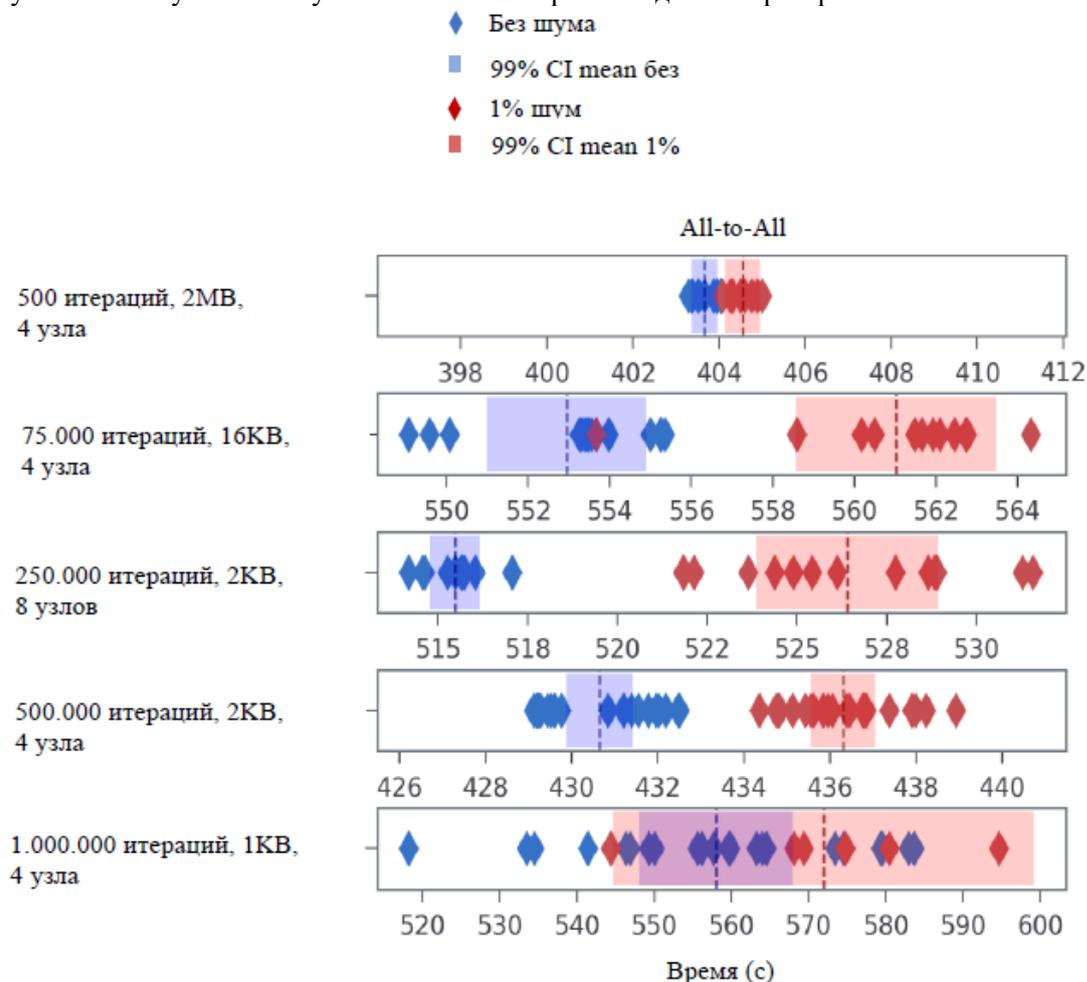
All-Reduce с 1.000.000 операций (рисунок 3, таблица 1) демонстрирует разброс во времени выполнения без шума, поэтому данная конфигурация не используется. Не исключено, что операция All-Reduce окажется пригодной в качестве детектора шума при должном подборе параметров запуска.

Для All-to-All с 500 итераций, 2 МВ, 4 узла на рисунке 4 видно, что время с шумом 1% слабо отличается от «чистого» времени. Поэтому в дальнейшем эта конфигурация операции All-to-All не использовалась.

75.000 операций All-to-All, 16 КВ, 4 узла (рисунок 4, таблица 2) можно считать пригодным инструментом для обнаружения шума.

Детекторы All-to-All, 250.000 итераций, 8 узлов и All-to-All, 500.000 итераций, 4 узла (рисунок 4, таблица 2) с размером сообщения 2 КВ для обоих случаев показывают наиболее стабильный результат, с хорошей возможностью обнаружения шума 1%.

В случае All-to-All 1.000.000 итераций, 1 КВ, 4 узла время «чистых» запусков оказалось сильно разбросанным. При таких параметрах инструмента уловить разницу между «чистыми» запусками и запусками с шумом 1% нельзя. Причины данного разброса неизвестны.



**Рис. 4.** Распределение времени для операции All-to-All

Непредсказуемые значения — выбросы — в ходе проведения исследования наблюдались для всех выбранных коллективных операций. Возможно, на появление разницы во времени между запусками влияют конкретные узлы, которые выделяются на суперкомпьютере для запуска задачи, задержки во взаимодействие между ними.

**Таблица 2.** Результаты запусков для I: All-to-All, 500 итераций, 2 МВ, 4 узла;  
 I: All-to-All, 75.000 итераций, 16КВ, 4 узла; III: All-to-All, 250.000 итераций, 2КВ, 8 узлов;  
 IV: All-to-All, 500.000 итераций, 2 КВ, 4 узла; V: All-to-All, 1.000.000 итераций, 1 КВ, 4 узла.

		I	II	III	IV	V
Чистые Запуски	Среднее время, с	403,62	552,94	515,88	430,65	557,99
	Стандартное Отклонение, с	0,29	2,16	1,63	1,28	14,34
	Количество Запусков	10	12	13	22	23
Запуски с шумом 1%	Среднее время, с	404,55	561,04	526,39	436,31	571,96
	Стандартное Отклонение, с	0,32	2,74	3,28	1,24	16,59
	Количество Запусков	8	12	15	22	6

### 3.6 Тестирование чувствительности

Выше были приведены результаты демонстрирующие способность разработанных программных средств улавливать шум модели агента мониторинга производительности в 1%. Ниже на графиках приведены результаты экспериментов, в которых уровень шума постепенно снижался до порога чувствительности (до верхней границы доверительного интервала).

Для тестирования были выбраны инструменты и их конфигурации, на которых шум 1% различим, то есть 99% доверительные интервалы времени без шума и с шумом 1% не пересекаются:

- All-to-All 500.000 итераций, 2 КВ, 4 узла — таблица 3
- All-to-All 250.000 итераций, 2 КВ, 8 узла — таблица 4
- All-to-All 75.000 итераций, 16 КВ, 4 узла — таблица 5
- Barrier 49.000.000 итераций, 4 узла — таблица 6

На рисунке 5 и в таблицах 3-6 в качестве значения шума при каждом из указанных уровней нагрузки представлено среднее арифметическое времени по всем соответствующим запускам в секундах.

По графикам на рисунке 5 видно, что при нагрузке в 0,5% шума для детекторов, использующих операции Barrier и All-to-All, отличаются время с шумом и без. При уровне шума 0,3% наблюдается достижение границы чувствительности трех из четырех рассмотренных конфигураций (для All-to-All 75.000 итераций, 16 КВ, 4 узла — вопрос открытый). Так как границы доверительных интервалов с шумом и без близки, при обнаружении шума 0,3% детекторы нужно использовать с осторожностью. В эксперименте All-to-All 250.000 итераций, 2 КВ, 8 узлов при уровне шума 0,05% наблюдается неожиданно большое время для такой нагрузки. Причина появления данного выброса не установлена.

**Таблица 4.** Результаты тестирования чувствительности All-to-All 250.000 ит., 2 КВ, 8 узлов

Уровень шума	Без шума	0,01%	0,05%	0,1%	0,3%	0,5%	0,7%	1%
Среднее значение, с	515,88	517,35	522,65	519,23	519,30	526,46	528,95	526,40
Стандартное отклонение, с	1,63	3,43	0,73	4,29	1,12	0,89	2,66	3,28
Количество запусков	13	5	5	5	5	5	4	15

**Таблица 3.** Результаты тестирования чувствительности All-to-All 500.000 ит., 2 КВ, 4 узла

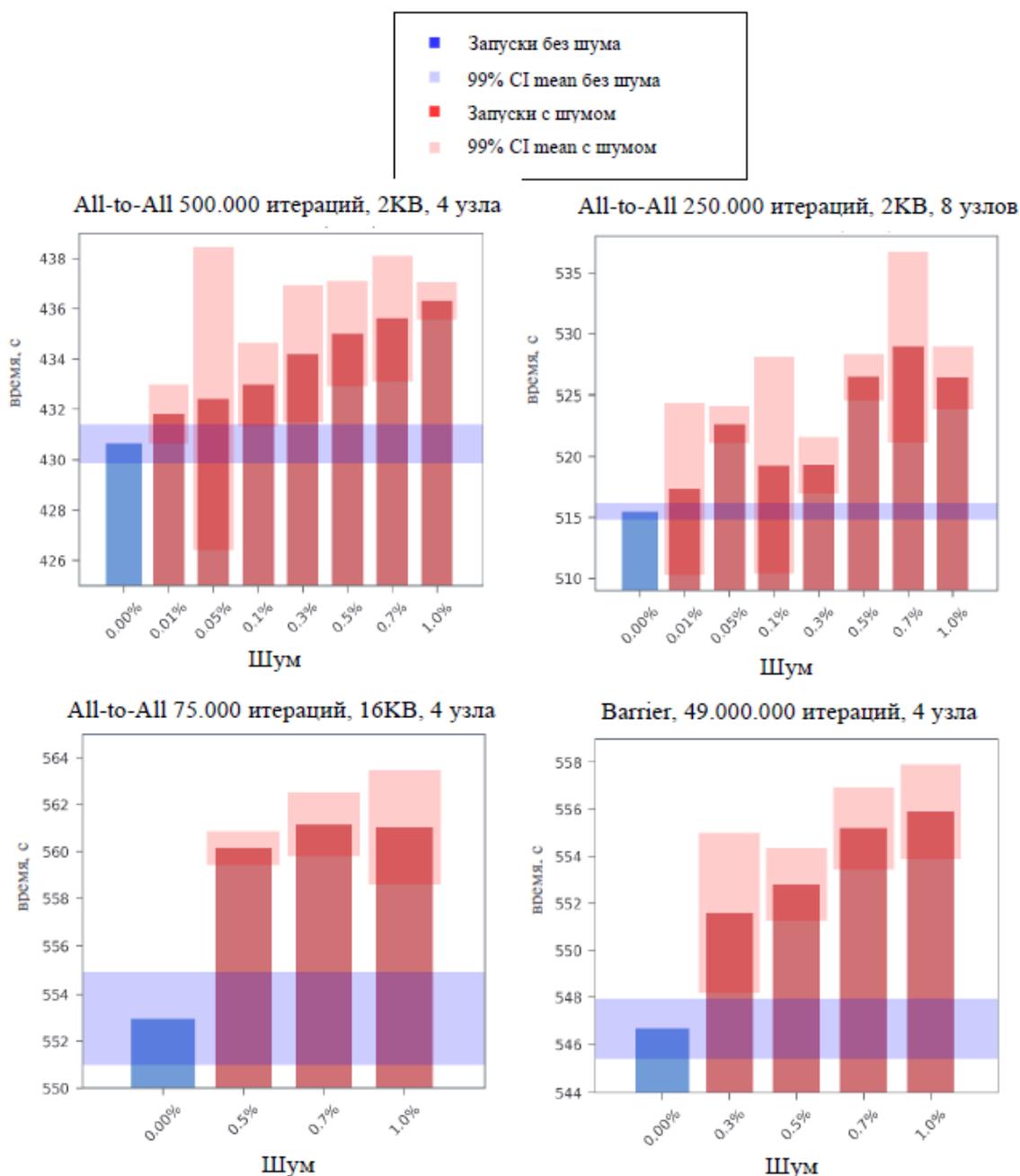
Уровень шума	Без шума	0,01%	0,05%	0,1%	0,3%	0,5%	0,7%	1%
Среднее значение, с	430,65	431,82	432,42	432,99	434,21	435,01	435,61	436,30
Стандартное отклонение, с	1,28	1,14	2,21	1,47	1,65	1,48	2,24	1,24
Количество запусков	22	10	3	10	7	8	10	22

**Таблица 6.** Результаты тестирования чувствительности All-to-All 75.000 ит., 16 KB, 4 узла

Уровень шума	Без шума	0,5%	0,7%	1%
Среднее значение, с	552,94	560,13	561,15	561,04
Стандартное отклонение, с	2,16	0,35	0,66	2,74
Количество запусков	12	5	5	12

**Таблица 5.** Результаты тестирования чувствительности Barrier, 49.000.000 ит., 4 узла

Уровень шума	Без шума	0,3%	0,5%	0,7%	1%
Среднее значение, с	546,66	551,58	552,80	555,19	555,90
Стандартное отклонение, с	1,03	1,66	0,75	0,85	2,09
Количество запусков	8,00	5,00	5,00	5,00	11



**Рис. 5.** Тестирование чувствительности

## 4. Результаты работы

Были разработаны два программных средства — источник шума, предназначенный для моделирования влияния работы агента системы мониторинга производительности, и детектор шума. В качестве детектора шума были опробованы программы с различными коллективными MPI операциями.

Для операции All-to-All проводились тесты с различной длиной сообщения. Оказалось, что наиболее стабильным из рассмотренных является детектор шума с длиной пересылаемого сообщения 2 КВ, протестированный на 4-х и 8-ми узлах суперкомпьютера. Детектор, использующий операцию Barrier, при большом числе итераций (время выполнения от 7 минут) также является чувствительным к шуму 1%. При использовании All-Reduce стабильная конфигурация для суперкомпьютера не была подобрана.

Для операций Barrier и All-to-All обнаружены граница чувствительности к нагрузке, вносимой источником шума. Это значение составляет 0,3% шума. Эти коллективные операции являются подходящими для обнаружения влияния системы мониторинга производительности в указанных конфигурациях.

## Литература

1. Sottile M.J., Minnich R.G. Supermon: a high-speed cluster monitoring system // Proceedings. IEEE International Conference on Cluster Computing. IEEE Comput. Soc, 2002. С. 39–46.
2. Mooney R., Schmidt K.P., Studham R.S. NWPerf: a system wide performance monitoring tool for large Linux clusters // 2004 IEEE International Conference on Cluster Computing (IEEE Cat. No.04EX935). IEEE, 2004. С. 379–389.
3. Adhianto L. и др. HPCTOOLKIT: tools for performance analysis of optimized parallel programs // Concurrency and Computation: Practice and Experience. 2010. Т. 22, № 6. С. 685–701.
4. Performance Co-Pilot [Электронный ресурс]. URL: <http://pcp.io/> (дата обращения: 31.03.2020).
5. Treibig J., Hager G., Wellein G. LIKWID: A Lightweight Performance-Oriented Tool Suite for x86 Multicore Environments // 2010 39th International Conference on Parallel Processing Workshops. IEEE, 2010. С. 207–216.
6. Rohl T. и др. LIKWID Monitoring Stack: A Flexible Framework Enabling Job Specific Performance monitoring for the masses // 2017 IEEE International Conference on Cluster Computing (CLUSTER). IEEE, 2017. Т. 2017–Septe. С. 781–784
7. Agelastos A. и др. The Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large Scale Computing Systems and Applications // SC14: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2014. С. 154–165.
8. Petrini F., Kerbyson D.J., Pakin S. The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q // ACM/IEEE SC 2003 Conference (SC'03). 2003.
9. Beckman P. и др. Benchmarking the effects of operating system interference on extreme-scale parallel machines // Cluster Computing. Springer US, 2008. Т. 11, № 1. С. 3–16.
10. Laguna I. и др. A large-scale study of MPI usage in open-source HPC applications // Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. New York, NY, USA: ACM, 2019. С. 1–14. DOI: 10.1145/3295500.3356176
11. Vl. Voevodin, A. Antonov, D. Nikitenko, P. Shvets, S. Sobolev, I. Sidorov, K. Stefanov, Vad. Voevodin, S. Zhumatiy: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. In Journal: Supercomputing Frontiers and Innovations, Vol.6, No.2 (2019). pp.4–11. DOI:10.14529/jsfi190201

12. Open MPI [Электронный ресурс]. URL: <https://www.open-mpi.org/> (дата обращения: 31.03.2020)
13. Hoefler T., Belli R. Scientific benchmarking of parallel computing systems: Twelve ways to tell the masses when reporting performance results // International Conference for High Performance Computing, Networking, Storage and Analysis, SC.: IEEE Computer Society, 2015. DOI: 10.1145/2807591.2807644

## Новая магистерская программа «Когнитивные системы»

Н.Ю. Золотых

Нижегородский гос. университет им. Н.И. Лобачевского

Когнитивные искусственные системы – это программные или программно-аппаратные комплексы, обнаруживающие в своем поведении элементы сознательной деятельности. Они способны решать задачи, решение которых ранее считалось прерогативой исключительно человека. Цель магистерской программы «Когнитивные системы» – подготовка разработчиков таких систем. В докладе представлена магистерская программа «Когнитивные системы», открытая в Нижегородском государственном университете в 2018 г.

*Ключевые слова:* когнитивные системы, искусственный интеллект, магистерская программа.

### 1. Введение

*Когниция* – это обработка информации нашим сознанием. Можно считать, что *когниция* является синонимом к слову *познание*. Таким образом, *когнитивные искусственные системы* – это программные и программно-аппаратные комплексы с элементами сознательного поведения. Они способны решать задачи, решение которых ранее считалось прерогативой исключительно человека, например, распознавание изображений, логический вывод, понимание и анализ текста, перевод с одного языка на другой, поддержка принятия решений, разработка стратегий в интеллектуальных играх (шахматы, го и др.).

В этом контексте термины *искусственные когнитивные системы* и *искусственный интеллект* можно считать синонимами. Далее слово «искусственный» в первом термине мы будем опускать.

Новый всплеск интереса к этой области объясняется достижениями последних лет в области разработки когнитивных систем, обусловленными успехами в машинном обучении, в первую очередь – в глубоком обучении. Необходима подготовка специалистов – разработчиков таких систем.

С 2018 г. в Нижегородском государственном университете им. Н.И. Лобачевского в магистратуре направления «Фундаментальная информатика и информационных технологии» действует магистерская программа «Когнитивные системы», направленная на подготовку таких специалистов.

Выпускники магистерской программы должны знать и понимать теоретический фундамент, на котором строятся когнитивные системы, знать и понимать алгоритмы машинного обучения и искусственного интеллекта, владеть конкретными инструментами решения, уметь строить из готовых блоков новые когнитивные системы.

В учебный план наряду с гуманитарными и общенаучными входят следующие основные дисциплины:

- алгоритмы и структуры данных;
- машинное обучение и анализ данных;
- прикладная математическая статистика;
- глубокое обучение;
- обучение с подкреплением

и дисциплины по выбору:

- профессиональный C++;
- программирование на скриптовых языках (Python);

- большие данные;
- интернет вещей;
- компьютерное зрение;
- обработка текстов на естественном языке;
- функциональное программирование;
- теория машинного обучения.

Учебные и научные практики и подготовка магистерской диссертации осуществляются в рамках выполнения реальных научных и производственных проектов.

Образовательную деятельность осуществляет профессорско-преподавательский состав, включающий 3 доктора наук и 10 кандидата наук, в том числе представителей индустрии (Интел, Мера, Harman, Marvell Technology Group). Программы дисциплин разработаны сотрудниками базовых кафедр. При преподавании ряда дисциплин (в частности, машинного обучения, глубокого обучения, обучения с подкреплением) используется материал, подготовленный преподавателями и разработчиками курсов Школы анализа данных (Яндекс).

Много времени в программе уделяется методам конструирования когнитивных систем на основе уже готовых блоков, что, возможно, является отличительной особенностью образовательной программы. Соответствующие навыки вырабатываются при выполнении курсовых и выпускных работ.

Далее приведем обзор программ некоторых ключевых дисциплин. При составлении обзора мы пользовались пакетом документов по презентуемой магистерской программе. Авторы программ дисциплин: «Машинное обучение и анализ данных» – проф. Н.Ю. Золотых, «Глубокое обучение» – доц. В.Д. Кустикова, «Прикладная математическая статистика» – проф. А.В. Зорин, «Компьютерное зрение» – доц. А.В. Бовырин, обработка естественных языков – проф. Н.Ю. Золотых, асс. О.В. Дурандин.

## 2. Машинное обучение и анализ данных

Ошеломляющие успехи последних лет в решении множества конкретных задач искусственного интеллекта обусловлены в первую очередь достижениями в области машинного обучения, особенно – глубокого обучения. Не будет ошибкой сказать, что машинное обучение стало основной технологией искусственного интеллекта. Более того, на машинное обучение возлагаются и дальнейшие надежды. Поэтому дисциплину «Машинное обучение и анализ данных» мы рассматриваем как одну из основных в презентуемой магистерской программе.

*Машинное обучение и анализ данных* – две близких, но, тем не менее, разных области. Основное различие, как мы полагаем, заключается в различии содержательных постановок рассматриваемых задач.

*Машинное обучение* занимается методами построения эффективных алгоритмов на основе примеров (обучающей выборки). Основная цель при этом – построение алгоритма решения той или иной сложной задачи с минимальной по возможности ошибкой (измеренной тем или иным способом). Главная цель *анализа данных* – в получении из них полезного интерпретируемого знания. В обеих областях работают с данными: в машинном обучении это примеры, на которых учится алгоритм, в анализе данных – это собранные каким-либо образом данные, из которых нужно получить интерпретируемое знание. В машинном обучении интерпретация уходит на второй план. Там главное – *обучить машину* хорошо (с небольшой ошибкой) решать задачу. Главное в анализе данных – *научить новому знанию человека*. Здесь величина ошибки, конечно же, важна, но уходит на второй план, а на первый план выходит интерпретируемость.

Несмотря на различия в содержательных постановках задач, методы в машинном обучении и анализе данных сильно пересекаются.

В результате освоения дисциплины студент должен знать основные подходы и алгоритмы машинного обучения и иметь навыки при решении конкретных практических задач, включая все этапы: предобработка данных, построение модели, оценка качества и др.

Перечислим основные рассматриваемые темы дисциплины:

1. Примеры и постановки задач обучения с учителем и обучения без учителя.
2. Основные метрики и эмпирические методы оценки качества решающих функций.
3. Метод  $k$  ближайших соседей.
4. Линейный метод наименьших квадратов. Борьба с переобучением. Регуляризация.
5. Наивный байесовский классификатор.
6. Линейный и квадратичный дискриминантный анализ.
7. Машина опорных векторов.
8. Логистическая регрессия.
9. Нейронные сети.
10. Основные идеи глубокого обучения.
11. Деревья решений.
12. Случайные леса и другие методы баггинга.
13. Градиентный бустинг деревьев решений.
14. Методы понижения размерности. Метод главных компонент.
15. Методы восстановления пропущенных значений.
16. Основные методы решения задач кластеризации.
17. Основы теории Вапника–Червоненкиса.

Лабораторные работы выполняются с использованием библиотек Scikit-Learn и Pandas на языке Python. Студенты в течение семестра на данных из конкретной предметной области должны отработать основные этапы работы с данными, включая их подготовку и анализ.

Рекомендуемая литература – [1, 2].

### 3. Глубокое обучение

Победа команды Krizhevsky, Sutskever, Hinton на соревновании ImageNet в 2012 г. ознаменовала начало этапа бурного развития и использования в компьютерном зрении, а затем и при решении многих других задач искусственного интеллекта, методов глубокого обучения. За прошедшее время глубокое обучение выросло из академического направления в технологию, оно используется в многочисленных прикладных областях.

В результате освоения дисциплины «Глубокое обучение» студент должен знать основные архитектуры применяемых глубоких сетей, алгоритмы их обучения, методы построения архитектур для решения тех или иных важных задач, а также уметь использовать эти знания для решения практических задач с использованием открытых библиотек глубокого обучения.

Перечислим основные темы, рассматриваемые в дисциплине:

1. Введение в глубокое обучение. Истоки и примеры задач.
2. Многослойные полностью связанные сети. Метод обратного распространения ошибки в таких сетях.
3. Обзор библиотек глубокого обучения Caffe, Torch, TensorFlow.
4. Сверточные нейронные сети. Различные слои (свертка, pooling, dropout, Local Contrast Normalization, Batch Normalization и др.). Функции активации и функции ошибки. Метод обратного распространения ошибки для сверточных нейронных сетей.
5. Визуализация состояний фильтров/выходов на промежуточных слоях сети. Классификация методов визуализации признаков. Открытые библиотеки для визуализации.
6. Рекуррентные нейронные сети и их развитие. Адаптация метода обратного распространения ошибки. Двухнаправленные рекуррентные нейронные сети. Глубокие двухнаправленные рекуррентные нейронные сети. Рекурсивные нейронные сети. Длинные рекуррентные нейронные сети с короткой памятью
7. Автокодировщик и стек автокодировщиков. Применение метода обратного распространения ошибки для обучения сети.

8. Разверточные нейронные сети. Ограниченная машина Больцмана. Глубокая машина Больцмана. Пример применения для начальной настройки параметров модели. Глубокая доверительная сеть.
9. Перенос обучения глубоких нейронных сетей.

Практические навыки закрепляются при выполнении следующих лабораторных работ:

1. Реализация метода обратного распространения ошибки для трехслойного перцептрона
2. Разработка полностью связанной нейронной сети с использованием одной из библиотек глубокого обучения для решения предложенной задачи. Проведение экспериментов с разным количеством скрытых слоев и числом скрытых элементов на каждом слое.
3. Разработка сверточной нейронной сети для решения предложенной задачи. Проведение экспериментов с разными конфигурациями сверточных нейронных сетей.
4. Визуализация фильтров, полученных на всех сверточных слоях нейронных сетей. Модификация параметров сетей и их конфигураций с целью повышения качества их работы.
5. Разработка рекуррентных нейронных сетей и их разновидностей для решения предложенной задачи. Проведение экспериментов с разными конфигурациями сетей.
6. Начальная настройка весов разработанных ранее нейронных сетей. Проведение экспериментов.
7. Применение переноса обучения для решения задачи, поставленной в ходе второй лабораторной работы. Проведение экспериментов с сетями, существующими для решения классических задач. Сбор результатов качества работы сетей с предварительной настройкой весов.

Рекомендуемая литература – [3].

#### 4. Прикладная математическая статистика

В дисциплине «Прикладная математическая статистика» затрагиваются разделы этой науки, наиболее часто использующиеся в современном анализе данных и машинном обучении: проверка гипотез, регрессионный анализ и др.

В результате освоения дисциплин студент должен знать и понимать математический аппарат для решения основных задач математической статистики и уметь применять их на практике.

Перечислим основные разделы дисциплины:

1. Знакомство с R. Виды данных. Генерация псевдослучайных чисел с различным законом распределения. Метод Монте-Карло.
2. Выборочное распределение и выборочные статистики. Ядерные оценки плотности.
3. Статистические гипотезы и виды ошибок. Критерии согласия. Проверка гипотез нормальности и экспоненциальности.
4. Доверительные интервалы. Методы получения точечных оценок.
5. Проверка гипотез о параметрах распределений. Лемма Неймана-Пирсона.
6.  $t$ -распределение и  $F$ -распределение
7. Непараметрические гипотезы однородности, независимости. Ранговые критерии
8. Регрессионный анализ.
9. Однофакторный дисперсионный анализ.

Лабораторные работы выполняются в среде статистических вычислений R. Список тем лабораторных работ:

1. Описательная статистика. Построение выборочных распределений и вычисление выборочных числовых характеристик в среде R
2. Оценка параметров распределений по методу максимального правдоподобия. Оценка параметров смесей
3. Байесовское оценивание параметров распределений.
4. Свойства оценок: распределение при нулевой гипотезе и при их нарушении
5. Проверка гипотез однородности: параметрические и непараметрические критерии
6. Критерии согласия Колмогорова – Смирнова и Пирсона.
7. Проверка нормальности: статистические критерии и графические методы
8. Регрессионная зависимость: модель Гаусса–Маркова, нелинейная регрессия.

Рекомендуемая литература – [4, 5].

## 5. Компьютерное зрение

Основная цель дисциплины «Компьютерное зрение» – познакомить студентов с основными алгоритмами решения задач компьютерного зрения и выработать навыки использования библиотек компьютерного зрения для решения практических задач.

Перечислим основные разделы дисциплины:

1. Методы получения и обработки изображений
  - a. Формирование изображений. Камера Обскура. Устройство современной цифровой камеры. Получение растра.
  - b. Устройство человеческого глаза. Типовое устройство системы компьютерного зрения.
  - c. Виды цифровых изображений. Основные форматы хранения.
  - d. Инструменты обработки бинарных изображений.
  - e. Обработка и низкоуровневый анализ полутоновых и цветных изображений.
  - f. Мультиспектральные изображения. Виды цветовых пространств. Методы улучшения цветных изображений. Методы сегментации цветных изображений.
2. Методы видеоанализа
  - a. Постановки задач видеонаблюдения.
  - b. Методы детектирование и оценки движения.
  - c. Обучение модели фона. Вычитание фона.
  - d. Постановка задачи слежения за объектом в видео потоке.
  - e. Численный метод поиска оптимального оптического потока.
  - f. Слежение за объектом с помощью алгоритма Meanshift.
  - g. Предсказание движения с помощью фильтра Калмана.
  - h. Детектирование подозрительных траекторий движения.
3. Методы поиска объектов на изображении
  - a. Постановка задачи поиска.
  - b. Обзор и классификация популярных методов локализации.
  - c. Метод скользящего окна.
  - d. Локализация особых точек изображения и вычисление вектора признаков методом SIFT.
  - e. Другие методы описания объекта (SURF, MSER).
  - f. Использование ключевых точек изображения для предсказания положения объекта.
  - g. Поиск шаблона с помощью решения двойственной задачи нахождения клики.
4. Методы локализации и распознавания лиц
  - a. Методы локализации лица.
  - b. Методы поиска элементов лица (глаза, нос, рот).
  - c. Методы распознавания лиц. Активные модели. Геометрическое сравнение. Поэлементное сравнение. Метод главных компонент. Использование оптического потока.

- d. Организация поиска в базе.
- 5. Численное описание, анализ и сравнение изображений
  - a. Постановка задачи поиска изображений. Практическая значимость.
  - b. Цветовые характеристики изображения.
  - c. Текстурные характеристики изображения.
  - d. Градиентные характеристики изображения.
  - e. Расстояние Хаусдорфа.
  - f. Различные численные методы сравнения изображений. Гистограммы. Корелограммы. LBP. Методы сравнения из стандарта MPEG-7.
  - g. Оптимальное хранение цифровой библиотеки. KD-деревья.
- 6. Моделирование визуально наблюдаемых процессов. Численные методы оценки модели
  - a. Примеры математического моделирования в задачах компьютерного зрения.
  - b. Метод наименьших квадратов. Преобразование Хафа.
  - c. Задача оценки модели движущегося человека.
  - d. Стохастические методы оптимизации модели. Метод фильтрации частиц.
- 7. Калибрация камер и стереозрение
  - a. Типы калибровки камер.
  - b. Модели камеры. Внутренние и внешние параметры камеры.
  - c. Обзор методов калибровки.
  - d. Стереозрение. Эпиполярная геометрия. 5. Восстановление структуры по движению.
  - e. Методы нахождения стереосоответствия.
- 8. Применение технического зрения в робототехнике
  - a. Планирование движений в условиях неопределённости.
  - b. Задача локализации робота.
  - c. Задача составления карты

#### Лабораторные работы:

1. Нахождение низкоуровневых характеристик изображения: градиенты, рёбра, угловые точки. Оптимальная бинаризация изображений методом Отцу. Векторизация и работа с контурами. Сегментация изображения.
2. Решение задачи автоматического отделения объекта от фона.
3. Решение задачи поиска объектов с помощью ключевых точек.
4. Численное решение задач классификации точек с помощью различных методов. Сравнительный анализ методов: ближайшего соседа, деревьев решений, машины опорных векторов, нейронных сетей с различной архитектурой. Экспериментирование с параметрами этих методов.
5. Численное решение задачи локализации лица на изображении.
6. Организация поиска изображений в базе различными способами.
7. Численный метод решения задачи оценки модели заданной кинематическим деревом.
8. Численный метод оптимальной калибровки камер и нахождения оптимального стерео-соответствия.

Рекомендуемая литература – [6–9].

## 6. Обработка текстов на естественном языке

Обработка текстов на естественном языке – еще одно важное приложение современных методов машинного обучения.

В результате изучения дисциплины студент должен знать основные математические модели и алгоритмы анализа текста на естественном языке, уметь работать с современными лин-

гвистическими ресурсами, понимать возможности и ограничения существующих методов автоматической обработки текстов, иметь практические навыки обработки текстов.

Перечислим основные темы дисциплины:

1. Введение в предмет. Основные задачи и методы. Автоматическая обработка текстов. Проблема неоднозначности в автоматической обработке текстов (лексическая, синтаксическая, семантическая неоднозначности, неоднозначности на уровне дискурса, на уровне прагматики и др.). Морфологическая разметка. Синтаксический разбор. Семантический анализ.
2. Компьютерная морфология. Морфологический анализ. Словарный и предиктивный морфологический анализ. Лексическая неоднозначность. Инструменты для морфологического анализа и методика их работы (АОТ, PyMorphy, MyStem, NLTK).
3. Языковая модель. Цепь Маркова,  $n$ -граммы. Задача определения части речи. Статистические методы определения части речи. Частеречевая разметка на базе скрытых Марковских цепей и алгоритм Витерби.
4. Исправление опечаток. Расстояние Левенштейна, расстояние Левенштейна–Дамерау. Подсчет расстояний Левенштейна. Инструментарий для исправления опечаток.
5. Морфологическая классификация естественных языков. Лингвистическая типология.
6. Синтаксический анализ в естественном языке. Синтаксическая неоднозначность. Подходы к описанию синтаксиса в естественном языке. Иерархия Хомского. Задача синтаксического разбора.
7. Грамматика зависимостей. Методы и алгоритмы синтаксического разбора в контексте грамматики зависимостей. Возможности и ограничения грамматики зависимостей.
8. Контекстно-свободные грамматики (КС-грамматики). Методы и алгоритмы синтаксического разбора в контексте КС-грамматик. Возможности и ограничения КС-грамматики. КС-грамматика как дополнение грамматики зависимостей.
9. Статистические методы синтаксического анализа. Оценка точности синтаксического анализа. Понятие проективности. SyntaxNet.
10. Семантический анализ. Формальные методы семантического анализа. Понятие онтологии. Модели представления знаний в компьютерной семантике. Онтологические ресурсы и компьютерные тезаурусы. Ресурсы WordNet, FrameNet. Тезаурусы для русского языка.
11. Дистрибутивная семантика. Word2Vec. Алгоритмы CBOW и Модель Skip-gram, GloVe. Исследование свойств предобученной модели Skip-gram модели обучение своей.

Рекомендуемая литература – [10–12].

## Литература

1. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer, 2016.
2. Уиттон Д., Хасты Т., Тибширани Р., Джеймс Г. Введение в статистическое обучение с примерами на языке R. ДМК Пресс, 2016.
3. Goodfellow I., Bengio Y., Courville A. Deep learning. MIT Press, 2017.
4. Федоткин М.А. Основы прикладной теории вероятностей и статистики. М.: Высшая школа, 2006.
5. Зорин А.В., Федоткин М.А. Введение в прикладной статистический анализ в пакете R: Учебно-методическое пособие. Нижний Новгород: ННГУ, 2010.
6. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. М.: Издательский дом «Вильямс», 2004.
7. Шапиро Л., Стокман Дж. Компьютерное зрение. М.: БИНОМ. Лаборатория знаний, 2006.

8. Kaehler A., Bradski G. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media, 2016.
9. Planche B., Andres E. Hands-On Computer Vision with TensorFlow 2: Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras. Packt Publishing, 2019.
10. Bird S., Klein E. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, 2009.
11. Eisenstein J. Introduction to Natural Language Processing. The MIT Press, 2019.
12. Lane H., Hapke H., Howard C. Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python. Manning Publications, 2019.

**От «Атомного проекта» к Всемирной Глобальной  
Научной Программе «Будущее Земли»:  
математика, компьютеринг, космос  
(в год 75-летия «Атомного проекта» к 110-летию со дня  
рождения незаменимого М.В.Келдыша посвящается)\***

Т.А. Сушкевич, С.А. Стрелков, С.В. Максакова

Институт прикладной математики им. М.В. Келдыша РАН

В год 75-летия «Атомного проекта» статья приурочена к 110-летию со дня рождения незаменимого М.В.Келдыша, с которым связан «золотой век» отечественной науки. В годы «ядерных» угроз в XX веке ученые спасли планету для человечества. «Атомный проект» явился главной движущей силой развития не только физики, математики и других наук, но впервые остро стояла необходимость создания больших математических счетных машин. В 40-ые годы США ввели в действие первую электронно-вычислительную машину. СССР ответил на этот вызов. Этот опыт сейчас очень востребован, поскольку угроза глобальных изменений на планете уже очевидна. Программа «Будущее Земли», инициатором которой является Международный научный совет, – естественный всеобъемлющий этап развития науки, технологий и общества, направленный на устойчивое развитие и обеспечение жизни на планете. Требуется консолидация всего мирового научного сообщества и мировых ресурсов суперкомпьютеров и информационных баз big data. Планета Земля – естественный пример динамической системы с нелинейными процессами, находящейся в непрерывных изменениях. Как в «Атомном» и «Космическом» проектах и при создании «Ракетно-ядерного щита», повышается роль математиков, «computer sciences» и космоса для реализации Программы, поскольку невозможны натурные эксперименты для исследования эволюции природной среды и климата планеты. В теоретических и прикладных исследованиях внедрился термин «Глобальная система», введенный академиком Н.Н.Моисеевым: необходимы анализ и синтез знаний о развитии планетарной цивилизации. Особую значимость приобретает проблема адекватной оценки роли и веса моделируемых подсистем в долгосрочной эволюции всей «Глобальной системы», в том числе связанных с радиационным полем Земли, объединяющим климат и экологию. В приоритете сопряженные прямые и обратные задачи – компьютерное моделирование радиационных процессов, прогностические расчеты радиационных характеристик и обработка огромных массивов данных глобального мониторинга и дистанционного зондирования Земли из космоса.

*Ключевые слова:* Атомный проект, М.В.Келдыш, математика, ЭВМ, математическое моделирование, компьютеринг, Будущее Земли

## 1. Введение

Вера в могучую силу разума, в высокое и благородное предназначение НАУКИ – вот источник неиссякаемых сил УЧЁНЫХ и ЛИДЕРОВ в XX-м веке! На том держалась почти 300-летняя ВЕЛИКАЯ Академия наук и особенно созданная в 1925 году Академия наук СССР (АН СССР) – форпост СССР в мире! Творчество, созидательный труд, культ научного подвига и всеобщая грамотность – вот основа достижений в XX-м веке! Кадры решали всё!

Глобальный вызов Конвенции «Повестка дня на XXI век» (Agenda 21) [1], принятой на Конференции ООН по окружающей среде и развитию, Рио-де-Жанейро, 3-14 июня 1992 года, Резолюции «Преобразование нашего мира: Повестка дня в области устойчивого развития на период до 2030 года» (Agenda 2030) [2], принятой Генеральной Ассамблеей ООН 25 сентября

---

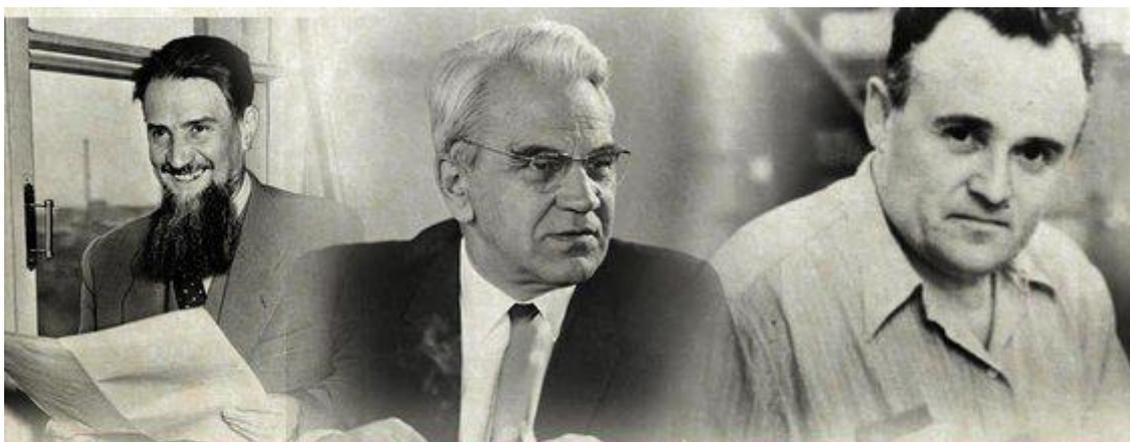
\* Исследование частично поддерживается грантами Российского фонда фундаментальных исследований (проекты 18-01-00609, 17-01-00220).

2015 года, и проект, не имеющий аналогов по масштабам и значимости для всего человечества в мировой истории, – это Всемирная Глобальная Научная Программа «Будущее Земли» (Программа) [3]. Формирование Программы началось в 2014 году прогрессивным научным сообществом для координации международных исследований по устойчивому развитию окружающей среды и общества по совместной инициативе Международного совета по науке (ICSU основан в 1931 году) и Международного научного совета по общественным наукам (ISSC основан в 1952 году). После объединения в 2018 году они переименованы в «Международный научный совет» (ISC) [4]. ISC – это неправительственная организация, представляющая 133 страны с уникальным глобальным членством, объединяющая 40 международных научных союзов и ассоциаций и более 140 национальных и региональных научных организаций, в том числе академии и исследовательские советы. В 2018 году к Программе присоединилась Россия в лице Национального Комитета при Президиуме Российской академии наук (РАН).

Главная задача Программы – координация и развитие новых трансдисциплинарных подходов к изучению взаимосвязи между процессами глобализации и изменениями окружающей среды, объединение усилий не только наук о Земле, но и «computer sciences» и «цифровизации». Поставлена задача усилить взаимодействие между фундаментальными исследованиями и практикой, ускорить передачу результатов научных разработок на политический уровень.

Двадцатый век в истории земной цивилизации – это век научно-технической революции (НТР), связанной с тремя эпохальными открытиями:

- проникновение в тайны и овладение ядерной энергией;
- покорение космического пространства и выход человека в космос;
- изобретение электронно-вычислительных машин (ЭВМ) и создание информационных технологий, которые стали движущей силой НТР и обеспечили успех «Атомного проекта» и «Космического проекта» и создания «Ракетно-ядерного щита» (далее – «Три проекта»).



### Три русских гения - Курчатов, Келдыш и Королев предопределили стратегический паритет

«Три К» – три русских гения, получивших высшее образование в вузах СССР, а не за рубежом, как ложно информируют, – Мстислав Всеволодович Келдыш (10.02.1911-24.06.1978) – Главный Теоретик космонавтики, Трижды Герой Социалистического Труда (1956, 1961, 1971); Сергей Павлович Королев (12.01.1907-14.01.1966) – Главный Конструктор космонавтики, Дважды Герой Социалистического Труда (1956, 1961); Игорь Васильевич Курчатов (12.01.1903-07.02.1960) – «Отец русской атомной бомбы», Трижды Герой Социалистического Труда (1949, 1951, 1954) – возглавили и совершили в XX-м веке в СССР научно-техническую революцию, и их имена навек вошли в историю мировой цивилизации человечества, СССР и России.

Эти академики – гордость ВЕЛИКОЙ Академии наук СССР! Эти Герои и прежде всего М.В.Келдыш – лучший Президент Академии наук (1961-1975) за всю её историю – подняли престиж НАУКИ и УЧЕНЫХ на такую недостижимую высоту, что АН СССР стала штабом нау-

ки в стране и УЧЕНЫХ СССР уважали во всем мире! Это благодаря их научным подвигам и заслугам был обеспечен стратегический паритет двух политических систем и военных блоков во главе с СССР и США и мир без глобальной войны.

Фундаментальные основы Программы для обеспечения устойчивого развития и существования человеческого общества и государств на планете заложены в XX-м веке: это беспрецедентный расцвет математики, изобретение компьютера, интернета и невиданных ранее информационных, коммуникационных и цифровых технологий, а также выход человека в космос, освоение космического пространства, разработки систем глобальных наблюдений и дистанционного зондирования Земли, объектов и процессов на базе «световых технологий» [5] при реализации «Трех проектов». Международная Радиационная Комиссия создана в 1896 году [6]. Ни в одной стране мира нет такого научного наследия и такой плеяды отечественных великих ученых, которые сформировались в период расцвета науки и техники в СССР под руководством АН СССР. Великие открытия – покорение атома и космоса и изобретение компьютера – базис научно-технической революции в середине XX-го века и международное сотрудничество в космосе связаны с именем математика-легенды Мстислава Всеволодовича Келдыша [7-13], который как Президент спас АН СССР от «реорганизации» в 1964 году [14].

«Историческая оптика» (термин введен В.С.Губаревым) помогает иногда сопоставить масштабы и значение происходящих в мире перемен. В 1975 году в СССР торжественно отмечалось 250-летие Академии наук, основанной Петром Первым в 1724 году, и 50-летие Академии Наук СССР, организованной И.В.Сталиным в 1925 году. Это был ПРАЗДНИК М.В.Келдыша, который отмечался широко, основательно и по заслугам достойно. По итогам этих торжественных мероприятий был издан сборник «250 лет Академии наук СССР. Документы и материалы юбилейных торжеств» [15]. Несомненно интересен раздел «Юбилейные научные сессии». Но сегодня актуально и важно обратить внимание на документы из раздела «Приветствия». Несмотря на весь свой официоз, лапидарность (лаконичная выразительность слога, стиля) и компактность, они дают представительный материал для сравнения с сегодняшним днем, с современным состоянием и Российской академии наук и мировой науки в целом. Эти приветствия позволяют еще раз задуматься о той роли, которую государство отводило и отводит науке.

## 2. «Атомный проект»: 1942-1945 гг. – начало

2020 год – это год 75-летия Великой Победы советского народа в Великой Отечественной войне 1941-1945 гг., но это и год 75-летия советского «Атомного проекта». В 2019 году к 70-летию успешного испытания первого советского «атомного заряда» 29 августа 1949 года раскритиковано множество документов и открыта для свободного открытого доступа Электронная библиотека «Атомный проект» [16]. Этот проект начинается с сов. секретного Постановления № 9887сс/оп Государственного Комитета Оборона (ГОКО) «**О Специальном комитете при ГОКО**» от **20 августа 1945 года**, подписанного Председателем Государственного Комитета Оборона И.Сталиным (Атомный проект: документы и материалы. Т. 2. Кн. 1. 1999. С. 11-14). В этом же Постановлении: «3. Для предварительного рассмотрения научных и технических вопросов, вносимых на обсуждение Специального комитета при ГОКО, рассмотрения планов научно-исследовательских работ и отчетов по ним, а также технических проектов сооружений, конструкций и установок по использованию внутриатомной энергии урана **создать при комитете Технический совет...**» и «4. Для непосредственного руководства научно-исследовательскими, проектными, конструкторскими организациями и промышленными предприятиями по использованию внутриатомной энергии урана и производству атомных бомб **организовать при СНК СССР Главное управление – «Первое главное управление при СНК СССР», подчинив его Специальному комитету при ГОКО.**»

Это был адекватный ответ на действия США, которые в августе 1945 года впервые применили атомное оружие, сбросив бомбы ужасающей силы на японские города Хиросима (6 августа урановая бомба) и Нагасаки (9 августа плутониевая бомба). Но когда Хиросима и Нагасаки фактически были сметены с лица Земли, потребовалось ускорение работ по созданию атомного оружия в СССР, тем более что поступали разведанные о возможном ядерном ударе и по Союзу Советских Социалистических Республик.

«Атомный проект» невозможно было реализовать без физиков-теоретиков и математиков, поскольку впервые познавали атом и покоряли ядерную энергию. Адекватные теоретико-расчетные исследования заменяли натурные эксперименты. Начиналась «Эпоха М.В.Келдыша» – «математика-легенды», который проложил дорогу «прикладным математикам» и компьютерам в «Атомном проекте», из которого вышел «Космический проект», а вместе это «Ракетно-ядерный щит» страны. **Впервые человечество столкнулось с вызовом: натурные эксперименты с атомными и ядерными бомбами и испытаниями взрывов и запусков ракет и спутников недопустимы, судьба планеты и человечества зависела от компьютеров и математиков.**

СССР проводил реальную миролюбивую политику и не спешил с разработкой средств для нанесения «ядерных ударов», но планомерно готовил научные кадры и развивал науку и технику. «Кузницей кадров» были МГУ и Бауманское училище в Москве, ЛГУ и Политехнический институт в Ленинграде. С 1946 года начал работу в виде физико-технического факультета МГУ знаменитый МФТИ, открытый в 1951 году (г. Долгопрудный). В Московском и Ленинградском университетах были созданы специальные отделения и новые кафедры. Были основаны Московский инженерно-физический институт (МИФИ) и его филиалы. К преподаванию привлекли ведущих ученых из научных институтов АН СССР, на их базе студенты и аспиранты проходили практику и выполняли дипломные и диссертационные работы.

Однако ключевым был **1942 год**. Разгар Великой Отечественной Войны. Только что защитили Москву. Сильнейшая в мире советская разведка (среди агентов и зарубежных ученых было немало сочувствующих СССР) поставляет достоверную информацию о работах в США, Англии, Германии, Франции. Бомбы ни у кого ещё не было, но работы уже велись. В настоящее время такие оригинальные достоверные материалы представлены в рассекреченных документах «Атомного проекта» с 1938 г. по 1957 г. [16]. Из писем агентов и ученых в адрес руководства СССР можно объективно уточнить состояние научных исследований по атомной и смежной тематике, а также о работах за рубежом. В теории советские ученые не отставали.

Отправной точкой истории отечественной атомной отрасли стало сов. секретное Распоряжение Государственного Комитета Оборона № 2352сс от **28 сентября 1942 года «Об организации работ по урану»**, подписанное Председателем Государственного Комитета Оборона И.Сталиным, с поручениями Академии Наук СССР (Атомный проект: документы и материалы. Т. 1. Ч. 1. 1998. С. 268-271):

*«Обязать Академию наук СССР (академик Иоффе) возобновить работы по исследованию осуществимости использования атомной энергии путем расщепления ядра урана и представить Государственному комитету обороны к 1 апреля 1943 года доклад о возможности создания урановой бомбы или уранового топлива.*

*Для этой цели:*

**1. Президиуму Академии наук СССР:**

*а) организовать при Академии наук специальную лабораторию атомного ядра...»*

Речь идет о создании в Москве на базе Ленинградского Физико-технического института АН СССР (основан в 1918 году; н. ФТИ им. А.Ф.Иоффе РАН) Лаборатории № 2 – Лаборатории измерительных приборов АН СССР (н. НИЦ «Курчатовский институт»).

«Работы по урану» советские ученые проводили на должном мировом уровне и были конкурентны, поскольку научный потенциал советских исследователей был достаточным и успешно развивался благодаря созданной в 1925 году Академии Наук СССР [7], роль которой была руководящей научно-организационной и консолидирующей, а также основанным после 1917 года академическим научным институтам (Радиевый институт, Физико-технический институт, Физико-математический институт и др.). Необходимо извлечь бесценные уроки из этих успешных проектов – они способствовали развитию теории и приложений математики, механики, физики, химии, наук о Земле, «computer sciences» и других.

Впервые за почти 100-тысячную историю человечества в XX-м веке был достигнут такой мощный прогресс, который на многие десятилетия и в XXI-м веке определил новый уклад – постиндустриальное информационное общество, и вместе с тем создано оружие уничтожения самой цивилизации! Советскому Союзу и всему человечеству планеты Земля повезло, что во главе этого прогресса стояли такие ученые-академики как М.В.Келдыш, С.П.Королев, И.В.Курчатов, которые своим талантом, умением, своими уникальными организационными

способностями лидеров и научными подвигами спасли планету от катастрофы и остановили безумную гонку вооружений, обеспечив безопасность страны, и тем самым поддерживая мирное сосуществование более 200 государств без глобальных войн на планете вот уже 75-ый год.

### **3. За математику, ЭВМ и расчеты отвечал незаменимый М.В.Келдыш**

И.В.Курчатов отвечал за «Атомный проект» и создание «изделий», С.П.Королев – за создание «ракет» для доставки «изделий» и космической группировки ПРО (противоракетной обороны), а М.В.Келдыш – за математику, вычислительную технику и расчеты в «Трех проектах».

Мстислав Всеволодович Келдыш – русский ГЕНИЙ, «Ломоносов XX-го века», аристократ, дворянин, интеллигент, крупнейший организатор советской науки и образования, математик-легенда – поднял престиж МАТЕМАТИКИ и КОМПЬЮТЕРОВ, заложив основы фантастического прогресса научно-технической революции XX-го века и нового постиндустриального информационного общества и «цифровизации» в XXI-ом веке.

Мстислав Всеволодович Келдыш - УЧИТЕЛЬ – ОБРАЗЕЦ СЛУЖЕНИЯ НАУКЕ и Родине, уникальная историческая личность ГРАЖДАНИНА и УЧЕНОГО, масштаб достижений и заслуг которого в XX-м веке никто в мировой цивилизации не смог и не сможет достичь, а тем более превзойти! Урок М.В.Келдыша: «свободу» понимал как «СВОБОДУ МЫСЛИ и ТВОРЧЕСТВА» – созидания! М.В.Келдыш – уникальный руководитель нового типа (академическая атмосфера общения и уважения)!

Мстислав Всеволодович Келдыш родился в Риге 10 февраля 1911 года в семье адъюнкт-профессора Рижского Политехнического института Всеволода Михайловича Келдыша, крупного инженера-строителя, «отца русского железобетона», академика и вице-президента Академии строительства и архитектуры. Нетрудно вычислить, что период его отрочества, юности, образования и формирования личности приходится на первые 15 лет советской власти, и совершенно очевидно, что нет никаких оснований подозревать советскую власть в любви к Келдышу, а Келдыша – в любви к ней. А.А.Ляпунов рассказывал, что Келдыша несколько раз пытались исключить из Московского университета за «непролетарское происхождение», и лишь усилиями его учителя М.А.Лаврентьева это удалось предотвратить. В подобных обстоятельствах многие ломаются, теряются, озлобляются и переносят свое резко отрицательное отношение к режиму и властям на Родину и народ в целом.

М.В.Келдыш принадлежит к другому типу людей. Образование, воспитание, врожденное чувство патриотизма сформировали в нем твердое убеждение: власть и Родина не тождественны, Родина у человека одна («запасных» нет), жизнь и шанс подарить людям свое творчество даются один раз и даются Богом, а не властями». К тому же, в конце 30-х годов возникла опасность очередной агрессии Запада – в этот раз со стороны германского фашизма. Все это определило жизненный выбор Келдыша: его научная деятельность была нацелена на развитие авиации и ракетной техники, математики и компьютеринга в реализации «Трех проектов».

Учитель М.В.Келдыша – это Михаил Алексеевич Лаврентьев (19.11.1900-15.10.1980), Герой Социалистического Труда (1967) за достижения созданного в 1957 году Академгородка. Оба из профессорских семей – воспитанные, культурные, просвещенные, высокообразованные и им было легко общаться, хотя Лаврентьев был на 10 лет старше. Дружили и сотрудничали до последних дней жизни М.В.Келдыша, но это тот случай, когда неамбициозный ученик превзошел амбициозного учителя! М.А.Лаврентьев и М.В.Келдыш – они были разные, но оба были интеллигентами и едины в том, что общенародное и государственное служение ставили выше личного интереса. И напрасно некоторые переписывают историю жизни и сотрудничества этих талантливых руководителей, которые стояли у истоков создания первых ЭВМ в СССР.

**В 1927 – 1931 гг.** Келдыш учился на математическом отделении физико-математического факультета МГУ. По соседству с «Лузитанией» находилась лаборатория Н.Е.Жуковским. Первый спецкурс молодой Келдыш выбрал сознательно – это гидромеханика, предугадывая возможность в ближайшем будущем приложения математических результатов в аэромеханике. В 20 лет выполнил дипломную работу под руководством М.А.Лаврентьева. В 30-х годах оборонная промышленность стала привлекать для решения новых проблем ученых разных специальностей и особенно математиков. Выпускник кафедры теории функции и функционального анализа М.В.Келдыш окончил МГУ с отличием по специальности «чистая математика» и по соб-

ственному желанию был направлен на работу в ЦАГИ, где проработал с 1931 по 1946 гг. и получил две Сталинские премии.

В 1934 году по указу И.В.Сталина АН СССР переводят из Ленинграда в Москву и на базе Физико-математического института создают Физический институт АН СССР (ФИАН) и Математический институт им. В.А.Стеклова АН СССР (МИАН). В МИАН отдел по теории конформных отображений и теории функций комплексных переменных возглавил М.А.Лаврентьев, который пригласил М.В.Келдыша на работу – по совместительству работал параллельно с ЦАГИ и поступил в аспирантуру МИАН (перешедшую в докторантуру). В МИАН под руководством М.А.Лаврентьева М.В.Келдышу в 1935 году присуждена ученая степень кандидата физико-математических наук, в 1936 году – кандидата технических наук и присвоено звание профессора по специальности «аэродинамика», в 1938 году – доктора физико-математических наук, тема диссертации: «О представлении рядами полиномов функций комплексного переменного и гармонических функций». В 1938 – 1953 гг. он профессор МГУ.

В апреле 1942 года М.В.Келдышу (совместно с Е.П.Гроссманом) присуждена Сталинская премия II степени за научные работы по предупреждению разрушений самолетов (флаттер). В июне 1943 года за выдающиеся заслуги в области научно-исследовательских работ в авиации М.В.Келдыш награжден первым Орденом Трудового Красного Знамени. 30 сентября 1943 года его избирают членом-корреспондентом АН СССР по Отделению физико-математических наук. В 1946 году М.В.Келдыш получает Сталинскую премию II степени за научные исследования в области теории и методов расчета автоколебаний самолетных конструкций (шимми). Математики до сих пор сопровождают эпитетом «красивая» любое упоминание об этой работе. В ней дано решение проблемы и предложены практические инженерные рекомендации. Это был **первый опыт приложений математики, который заметили ученые и руководители страны.**

На Общем собрании 30 ноября 1946 года М.В.Келдыш (35 лет) избран действительным членом Академии наук в один день с А.М.Лаврентьевым (46 лет) и Л.Д.Ландау (38 лет). После избрания в академики в Отделение технических наук по специальности «математика» М.В.Келдыш стал признанным главным специалистом и лидером по «прикладной математике».

Узнав о таком признании заслуг М.В.Келдыша со стороны академиков, **руководство страны приняло судьбоносное решение:** если бы М.В.Келдыш перешел «на основную работу в лабораторию академика Семенова», он бы не реализовался и не состоялся ни как Главный Теоретик космонавтики ни как Президент АН СССР! И.В.Сталин назначает М.В.Келдыша (Приказ № 462 от 02 декабря 1946 года) Начальником Реактивного научно-исследовательского института (НИИ-1 МАП – Министерство авиационной промышленности) и начинается сотрудничество с С.П.Королевым, которого в 1946 году направили в головной институт по ракетной технике НИИ-88 заведовать Отделом № 3 – основой будущего знаменитого ОКБ-1.

Это был первый шаг М.В.Келдыша к «Космическому проекту» и ко второй звезде Героя Социалистического труда в 1961 году «за обеспечение полета первого человека в космос». А третью звезду получил в 1971 году за успехи отечественной науки и Академии наук СССР как лучший Президент за всю историю Академии наук.

И начиналось это восхождение М.В.Келдыша в 1946 году, когда в «борьбу» за «молодого талантливого математика» вступили руководители – организаторы работ по «Атомному проекту» (Б.Л.Ванников, А.П.Завенягин, И.В.Курчатов, Н.Н.Семенов, Ю.Б.Харитон, Я.Б.Зельдович), а руководство авиационной промышленности не желало с ним расставаться.

**Николай Николаевич Семенов** (15.04.1896-25.09.1986) – академик с 29.03.1932 года, Дважды Герой Социалистического Труда (1966, 1976), Нобелевская премия по химии в 1956 году, директор Института химической физики АН СССР (ИХФ): *«Наша советская математика является самой сильной и мире. Эту силу мы должны использовать – это наш козырь. Проф. Келдыш – сильнейший математик, находящийся в самом творческом возрасте и активно желающий сосредоточить все свои силы на новой проблеме. Мне кажется, что этому его желанию препятствовать нельзя. Я придаю огромное значение привлечению его к новой проблеме. Как только он овладеет новой областью, создастся возможность втягивания в проблему всех основных математических сил.»*

Даже в научных кругах многим не было известно, какую ключевую роль в судьбе М.В.Келдыша сыграл **Иван Матвеевич Виноградов** (14.09.1891-20.03.1983) – академик с

12.01.1929 года, Дважды Герой Социалистического Труда (1945, 1971), первый директор Математического института им. В.А.Стеклова АН СССР. И.М.Виноградов [10]:

*«Я считаю неправильным, когда математика обособлена от приложений. Природа не знает искусственных постановок, а математика – знает. Келдыш работал в ЦАГИ и задачи брал из жизни. В 1944 г. через вице-президента Г.М.Кржижановского снова организовали отдел механики (во главе с М.В.Келдышем. – Примеч. составителей). Бардин был против этой организации. Я ему говорил: "Вот мы воспитали, а вы взяли, а сейчас мы у вас ничего не берем". Келдыш тогда взял к себе в отдел Дородницына и Седова, потом Охоцимского. При отделе стал работать научный семинар, объединивший всех московских аэродинамиков.*

*Вскоре после войны пришли ко мне Ю.Б.Харитон и другие физики. Просили порекомендовать математика, который мог бы поставить расчеты по атомной тематике. Я им порекомендовал взять Келдыша, – он в любом приложении математики способен разобратся лучше всякого. Келдыш им понравился. Прикладной математикой у нас в институте всегда много занимались, особенно много делали во время войны. Вот Келдыш и организовал к осени 1946 г. расчетное бюро, сначала из старых сотрудников, а потом туда пришло много молодежи.*

*В конце 1946 г. были выборы в Академию, по отделению технических наук было место, а хороших кандидатур у них не было. Я позвонил академику-секретарю в полдвенадцатого ночи: выберите моего Келдыша. Б.А.Введенский обрадовался: "Это очень важно, что он по механике". Келдыша в академии выбрали и послали руководить институтом по его ракетно-авиационной специальности. Снова стал он моим заместителем уже в 1949 г., и с тех пор вел всю прикладную математику.»*

Увидев фамилию М.В.Келдыша в письмах с просьбой о переводе М.В.Келдыша на работы по «Атомному проекту» после его успехов в авиации, И.В.Сталин воскликнул: «Опять этот Келдыш!». В просьбах было отказано, но нашли компромиссное решение. И это была судьба! В том же 1946 году М.В.Келдыша избрали академиком АН СССР и нашли ему работу, на которой он смог реализовать себя в большей мере. После ЦАГИ М.В.Келдыш одновременно работал руководителем РНИИ, зам. директора Математического института им. В.А.Стеклова АН СССР, затем директором ОПМ МИАН СССР (секретный «Институт Келдыша» – Отделение прикладной математики Математического института им. В.А.Стеклова АН СССР, созданный в 1953 году), профессором МГУ и МФТИ. А самое важное – он стал **главным, кто отвечал за МАТЕМАТИКУ, РАСЧЕТЫ и ЭВМ в СССР!**

#### **4. Почему мы были первыми в космосе?**

А.П.Александров. Речь на похоронах М.В.Келдыша, Красная площадь. 29.06.1978 г. [10]:

*«Когда перед нашей страной возникла необходимость решения атомной проблемы, под руководством и при непосредственном участии М.В.Келдыша были созданы сложнейшие методы расчетов для этой новой области, и под его влиянием появилась необходимая вычислительная техника. "Три К" – Курчатов, Келдыш и Королев – были учеными и организаторами научных разработок, позволивших создать ракетно-атомный щит нашей Родины, который отрезвил агрессоров и создал предпосылки к последовательной политике мира, проводящейся нашим государством.*

*Именно деятельность М.В.Келдыша превратила Академию в главный штаб науки. Для всей его организаторской деятельности было характерно правильное определение стратегии научных исследований, определение соотношения между фундаментальными и прикладными направлениями.*

*Мстислав Всеволодович Келдыш в науке был чрезвычайно требовательным, он совершенно не переносил верхоглядства и рекламы. Его служение науке было высоко принципиальным и самоотверженным. Только тяжелая болезнь помешала ему оставаться на посту президента Академии наук СССР до конца своей жизни.*

*Как человек, Мстислав Всеволодович поражал необычайной демократичностью, гибкостью ума, желанием и умением глубоко понять собеседника, вникнуть в его человеческую сущность, помочь и в научных, и в жизненных трудностях.»*

Из статьи В.С.Губарева «Три звезды президента»:

*«Иногда мне кажется, что гении – очень простые и доступные люди. Они всегда рядом, с ними всегда можно посоветоваться и найти выход из любого, даже самого трудного положения. Гении помогают нам по достоинству оценивать любые события и всех людей – от рядовых до избранных. Вот почему некоторых гениев власти любят и лелеют, а об иных стараются забыть. Однако есть люди, неподвластные сиюминутности, они вошли в историю Родины навсегда, потому что их дела бессмертны, как и народ, их породивший.*

*В ряду тех, коими славна наша Отчизна, особое место занимает Мстислав Всеволодович Келдыш. Келдыш – гений, и никто не может оспаривать это, а потому остается только изучать его труды, ставшие классикой, да подсчитывать все увеличивающееся число его учеников, так как математическая "школа Келдыша" не умерла вместе со своим создателем, а была и есть в том самом институте, который теперь носит его имя.*

*Но был и другой Келдыш... Он открывался редко, чаще всего его красивое лицо, окаймленное благородной сединой, оставалось суровым, непроницаемым, будто хозяин его недоступен лишь избранным... "Эй, как у вас там дела на Олимпе?" – хочется крикнуть таким людям...»*

**1951 год – ключевой в жизни М.В.Келдыша.**

Из Постановления СМ СССР № 1552-774сс/оп от 9 мая 1951 года «О работах по РДС-6Т» (Атомный проект: документы и материалы. Т. 3. Кн. 1. 2008. С. 396-403.):

4. Обязать Первое главное управление при Совете Министров СССР (гг. Ванникова, Завенягина) организовать **в составе Научно-технического совета математическую секцию** (секцию № 7) для научного руководства по разработке конструкций, быстродействующих вычислительных машин, а также методов их эксплуатации в составе:

<b>академик Келдыш М.В.</b>	—	<b>председатель секции</b>
академик Петровский И.Г.	—	член секции
академик Соболев С.Л.	—	- " -
член-корреспондент АН СССР Боголюбов Н.Н.	—	- " -
член-корреспондент АН СССР Тихонов А.Н.	—	- " -
академик Лаврентьев М.А.	—	член секции (по вопросам вычислительных машин)
член-корреспондент АН СССР Лебедев С.А.	—	- " -
инженер Базилевский Ю.Я.	—	- " -
инженер Лесечко М.А.	—	- " -

Возложить на секцию № 7 рассмотрение планов научно-исследовательских, экспериментальных и проектных работ, а также проектов математических машин и планов работ организаций, выполняющих расчетные работы по тематике Первого главного управления при Совете Министров СССР.

Председатель Совета Министров Союза ССР И. Сталин

## 5. Заключение

Уровень мобилизации профессиональных кадров, прежде всего УЧЕНЫХ, ИНЖЕНЕРОВ, КОНСТРУКТОРОВ, беспрецедентный уровень организации работы многочисленных коллективов и нескольких ведомств, мощнейший интеллект образованного советского народа и гениальность творцов до сих пор поражают специалистов. Это был ТРУДОВОЙ ПОДВИГ талантливого народа, который только что победил фашизм и вновь встал на защиту своей Родины.

10 февраля 2020 года, как всегда, возложили цветы к захоронению М.В.Келдыша в Кремлевской стене на Красной площади. Академик М.Я.Маров, который 20 лет работал с М.В.Келдышем и был его референтом, вспомнил, как в Кремле проходили «субботники» у Министра обороны маршала Дмитрия Федоровича Устинова (30.10.1908-20.12.1984). В непринужденной обстановке каждый из присутствующих мог высказаться. Обсуждение растягивалось на весь день. М.В.Келдыш обычно сидел молча и только иногда задавал короткие уточняющие вопросы. Для завершения встречи Д.Ф.Устинов обращался к М.В.Келдышу (по име-

ни): «Мстислав, а что ты думаешь?» М.В.Келдыш четко и очень конкретно сформулировал итоги совещания и министр произносил вердикт: «Вот так и запишем в решении».

Гении не нуждаются в почитании, память о них нужна нам, живущим, и тем, кто придет нам на смену. Когда рвется ниточка памяти, протянутая из прошлого в будущее, нация деградирует и погибает. Помним ли мы об этом?!

## Литература

1. Повестка дня на XXI век (Agenda 21) // Доклад Конференции Организации Объединенных Наций по окружающей среде и развитию Рио-де-Жанейро, 3-14 июня 1992 года. Том 1. Резолюции, принятые на Конференции. Нью-Йорк: ООН, 1993. 8-511 с. URL: <https://undocs.org/ru/A/CONF.151/26/Rev.1%28Vol.I%29> (дата обращения 15.04.2020).
2. Преобразование нашего мира: Повестка дня в области устойчивого развития на период до 2030 года (Agenda 2030) // Резолюция, принятая Генеральной Ассамблеей ООН 25 сентября 2015 года. URL: <https://undocs.org/ru/A/RES/70/1> (дата обращения 15.04.2020).
3. Future Earth. World Global Research Projects. URL: <http://futureearth.org>; <https://unfccc.int/news/future-earth-research-for-global-sustainability> (дата обращения 15.04.2020).
4. The International Science Council (ISC). URL: <https://council.science/> (дата обращения 15.04.2020).
5. Сушкевич Т.А. Математические модели переноса излучения. Москва: БИНОМ. Лаборатория знаний, 2005. 661 с.
6. International Radiation Commissions 1896 to 2008: Research into Atmospheric Radiation from IMO to IAMAS / Compiled by Hans-Jürgen Bolle from IRC documents as well as publications and notes by Fritz Möller and Julius London. Oberpfaffenhofen, Germany, May 2008. 138 p.
7. Президент Академии наук СССР М.В.Келдыш: 100 лет со дня рождения // Архив Академии наук. 2011. 49 с. URL: [http://www.arran.ru/data/collections/col7\\_.pdf](http://www.arran.ru/data/collections/col7_.pdf) (дата обращения 15.04.2020).
8. Мстислав Всеволодович Келдыш. К 100-летию со дня рождения. URL: <http://келдыш.рф>; <https://keldysh.ru/memory/keldysh/index.htm> (дата обращения 15.04.2020).
9. 4 октября 1957 года начало космической эры. Первая космическая // Сборник статей, посвященных пятидесятилетию юбилею запуска Первого искусственного спутника Земли. М.: ИКИ РАН, «Регион Инвест», 2007. 169 с.
10. 10 Келдыш М.В. Творческий портрет по воспоминаниям современников. М.: Наука, 2001. 416 с.
11. Мстислав Келдыш // «Великие умы России» под ред. В.С.Губарева. Выпуск 2. М.: Издательский дом «Комсомольская правда», 2016. 96 с.
12. Сушкевич Т.А. Главный Теоретик М.В. Келдыш и Главный Конструктор космонавтики С.П. Королев – покорители космоса // Современные проблемы дистанционного зондирования Земли из космоса. 2011. Т. 8. № 1. С. 9–25.
13. Сушкевич Т.А. М.В. Келдыш организатор международного сотрудничества в космосе и первой советско-американской Программы «Союз-Аполлон» (ЭПАС) // Современные проблемы дистанционного зондирования Земли из космоса. 2011. Т. 8. № 4. С. 9–22.
14. Афиани В.Ю., Илизаров С.С. Вестник истории естествознания и техники. 1999. № 1. С. 167–173.
15. 250 лет Академии наук СССР (1724-1974). Документы и материалы юбилейных торжеств. М.: Наука, 1977. 585 с.
16. История Росатома. Атомный проект СССР. Электронная библиотека URL: <http://elib.biblioatom.ru/sections/0201/> (дата обращения 15.04.2020).
17. Афендикова Н.Г. О роли М.В.Келдыша в некоторые ключевые моменты становления отечественной вычислительной техники // Препринты ИПМ им. М.В.Келдыша. Москва. 2017. № 58. 12 с. DOI: 10.20948/prepr-2017-58.
18. Езерова Г.Н., Луховицкая Э.С. К вопросу об истории информатики в России // Препринты ИПМ им. М.В.Келдыша. Москва. 2012. № 49. 12 с. URL: <http://keldysh.ru/papers/2012/prep2012-49.pdf> (дата обращения 15.04.2020).

## Сравнение производительности пакетов симуляции квантовых вычислений QuEST и Intel-QS\*

А.В. Линев<sup>1</sup>, П.Е. Ведруков<sup>1</sup>, Д.С. Куландин<sup>1</sup>, И.Б. Мееров<sup>1</sup>, С.В. Денисов<sup>1,2</sup>

<sup>1</sup>Нижегородский государственный университет им. Н. И. Лобачевского,  
<sup>2</sup>Oslo Metropolitan University

В ближайшем будущем появятся квантовые компьютеры, пригодные для практического использования. Симуляторы квантовых вычислений помогают разрабатывать и отлаживать квантовые алгоритмы, способствуют ко-дизайну при разработке квантовых архитектур. Для планирования вычислительных экспериментов необходимо представлять возможности симуляторов по моделированию вычислений на доступном оборудовании. Мы представляем результаты экспериментов по оценке возможных размеров симулируемой системы (числа кубитов), производительности и масштабируемости на общей и распределенной памяти пакетов QuEST и Intel-QS для вычислительных систем различного уровня.

*Ключевые слова:* вычислительная квантовая физика, квантовые алгоритмы, высокопроизводительные вычисления, QuEST, Intel-QS.

### 1. Введение

Важной составляющей разработки и анализа квантовых алгоритмов [1] является симуляция их выполнения на классических компьютерах [2]. Разработаны десятки приложений [3], которые позволяют описывать квантовые схемы с использованием специализированных языков квантовых вычислений [4-7], выполнять симуляцию работы квантового алгоритма [8-11] и даже поддерживать полный цикл разработки, включающий описание алгоритма, его оптимизацию, отображение на конкретную архитектуру квантового компьютера, верификацию, симуляцию работы полученной квантовой схемы и оценку ее производительности [12].

Симуляция возможна только для квантовых систем, состоящих из малого числа кубитов. Даже использование суперкомпьютеров позволяет симулировать работу квантовых систем, состоящих всего лишь из 38 [8], 42 [9], 45 [13], 49 [14], возможно, 53 [15], а в специальных случаях – до 64 кубитов [16], в то время как Google уже представляет результаты экспериментов на реальном квантовом компьютере общего назначения из 53 кубитов [17], анонсирует квантовый процессор из 72 кубитов [18], а D-Wave systems – специализированную систему из 2000 кубитов, реализующую алгоритм квантового отжига [19].

Несмотря на различную демонстрируемую производительность [20], разнообразие приложений позволяет выбрать наиболее подходящее для конкретной задачи и имеющихся вычислительных ресурсов. При выборе программных средств моделирования квантовых вычислений мы провели ряд экспериментов для определения производительности различных пакетов по выполнению симуляции работы квантовой схемы на идеальном квантовом компьютере [2] на системах различного класса – от настольного компьютера до вычислительного кластера. Мы рассматривали пакеты, которые, с одной стороны, позволяют оценить потребление ресурсов при выполнении отдельных шагов моделирования, с другой стороны – имеют параллельные версии для общей и распределенной памяти. Множество пакетов не удовлетворяют одному из данных требований (IBM Qiskit, LIQUI), ProjectQ, DG-M и т.д.) или недоступны для свободного использования [21]. Мы выбрали для анализа два пакета, удовлетворяющие нашим требованиям и положительно характеризующиеся в публикациях – QuEST и Intel-QS. Полученные результаты будут использованы для формирования набора инструментов, которые будут использовать-

---

\* Исследования выполнены при поддержке гранта РФФ № 19-72-20086 с использованием вычислительных ресурсов СК Intel Endeavor.

ся для разработки и анализа квантовых алгоритмов и обучения студентов и аспирантов ННГУ им. Н. И. Лобачевского.

Работа построена следующим образом. В разделе 2 приведено описание тестовой задачи. В разделе 3 – описание проведенных экспериментов. Раздел 4 содержит характеристики вычислительных систем. Раздел 5 – результаты экспериментов и комментарии к ним. Обобщение результатов представлено в Разделе 6.

## 2. Тестовая задача

Квантовый компьютер хранит информацию в виде кубитов, которые можно считать квантовой версией битов. Базовые операции над кубитами включают квантовые гейты – аналоги классических битовых операций – и операцию измерения кубита.

Мы будем рассматривать симуляцию идеального квантового компьютера, в котором отсутствуют шумы и декогеренция, присущие реальным устройствам. В этом случае состояние квантового компьютера с  $N$  кубитами описывается квантовым состоянием (также называемым "волновой функцией") системы кубитов, представленным в виде вектора комплексных чисел размером  $2^N$  ( $a_i$ ),  $i \in \{0, 1\}^N$ ,  $a_i \in \mathbb{C}$ , удовлетворяющего условию нормировки  $\sum |a_i|^2 = 1$ .

Действие квантового гейта в общем случае вызывает изменение всех  $2^N$  элементов вектора, согласно природе физического процесса описывается унитарным преобразованием и может быть представлено унитарной матрицей. Квантовые компьютеры реализуют набор базовых гейтов, управляющих состоянием одного-трех кубитов и представляемых сильно разреженной матрицей преобразования системы в целом (комбинация кронекеровских произведений матрицы воздействия на кубиты и единичных матриц). Для повышения производительности симуляция квантовых гейтов обычно выполняется не универсальным умножением разреженной матрицы на плотный вектор, а реализацией специализированного алгоритма, уникального для каждого типа квантового гейта.

В тестовом примере мы использовали однокубитовый гейт преобразования Адамара ( $H$ ), двухкубитовый гейт контролируемого отрицания ( $C\text{-NOT}$ ) и вычисление вероятности измерения нулевого кубита в состояниях  $|0\rangle$  и  $|1\rangle$  ( $P_0$  и  $P_1$  соответственно), которые описываются следующими матрицами и формулами:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, P_0 = \sum_{i_1, \dots, i_{N-1}} a_{0, i_1, \dots, i_{N-1}}, P_1 = \sum_{i_1, \dots, i_{N-1}} a_{1, i_1, \dots, i_{N-1}}.$$

Первый из них воздействует на один кубит и требует вычислений с использованием и обновлением всех элементов квантового состояния, второй – воздействует на два кубита и не предполагает выполнения вычислений. Для выполнения одной операции Адамара для  $N$ -кубитной системы требуется выполнение  $2^N$  умножений и  $2^N$  сложений комплексных чисел, т.е.  $8 \cdot 2^N$  FLOP. При этом требуется считать из памяти  $16 \cdot 2^N$  байт и записать в память  $16 \cdot 2^N$  байт (например, для  $N=30$  и 50 операций – 400 GFLOP и 32 GB). Для выполнения одной операции контролируемого отрицания требуется выполнить 0 FLOP, считать из памяти  $8 \cdot 2^N$  байт и записать  $8 \cdot 2^N$  байт (для  $N=30$  и 50 операций – 0 GFLOP и 16 GB).

При выполнении экспериментов производилась симуляция выполнения на системе из  $N$  кубитов квантовой схемы, состоящей из 50 операций Адамара и 50 операций контролируемого отрицания. Рассматривались следующие этапы симуляции:

- инициализация программного окружения ПО симуляции;
- создание системы из  $N$  кубитов;
- последовательное применение оператора Адамара к кубитам  $0, 1, \dots, N-1, 0, 1, \dots$  (далее по циклу); общее число операций – 50;
- последовательное применение оператора  $C\text{-NOT}$  к парам кубитов ( $control=0, target=1$ ),  $(1,2), \dots, (N-2, N-1), (N-1,0), (0,1), \dots$  (далее по циклу); общее число операций – 50;
- вычисление вероятностей нахождения нулевого кубита в состояниях  $|0\rangle$  и  $|1\rangle$ ;
- удаление системы кубитов;
- завершение программного окружения ПО симуляции.

Оценивались времена работы и потребление оперативной памяти на всех этапах.

### 3. Описание экспериментов

Для оценки характеристик программных средств симуляции работы квантовой схемы мы использовали 3 реализации:

- QuEST – симулятор квантовых компьютеров с поддержкой параллельного выполнения на общей и распределенной памяти [22];
- Intel Quantum Simulator – симулятор, оптимизированный для многоядерных архитектур и вычислительных кластеров [23];
- "наивная" реализация, выполненная нами (не использует векторизацию, оптимизацию использования кеша и т.д.); используется для оценки ускорения качественных реализаций.

Все пакеты компилировались и запускались с параметрами по умолчанию.

Проводились следующие виды экспериментов.

1. Определение зависимости времени работы и объема используемой оперативной памяти от числа кубитов. Выполнялись тесты для числа кубитов от 2 до максимально возможного на вычислительной системе. На многоядерных системах использовались все ядра. Измерялось время работы каждого этапа и объем памяти, используемый процессом в конце этапа.

2. Масштабируемость на общей памяти. Использовалось 30 кубитов и число потоков OpenMP от 1 до максимально возможного на вычислительной системе. Измерялось время работы каждого этапа и используемый объем памяти.

3. Масштабируемость на распределенной памяти. Использовалось 30 кубитов и число MPI-процессов от 1 до максимально возможного на вычислительной системе. Измерялось время работы каждого этапа и объем памяти, используемый одним процессом в конце этапа.

На каждой вычислительной системе проводились все виды экспериментов, которые имело смысл на ней проводить.

### 4. Вычислительные системы

Для оценки возможности использования программных средств симуляции работы квантовой схемы мы использовали 3 вычислительные системы, существенно отличающиеся с точки зрения производительности.

#### 1. Виртуальная машина (VM), запущенная на рабочей станции.

- Конфигурация: Intel® Core™ i3-7100 (Kaby Lake), 3.90 GHz, 1 ядро, 1 GB DDR4-2132 (1066 MHz).
- Производительность: 62,4 GFLOPS/ядро, 17 064 MB/s \* 2 канала.
- Интерконнект: предоставляемый средством виртуализации.
- Операционная система: CentOS 7.4.1708.
- Компилятор: gcc 4.8.5-39, gcc 7.3.1-5.
- MPI: mpich-3.0.

#### 2. Dell PowerEdge R815 (64-ядерный сервер).

- Конфигурация: 4xAMD Opteron 6366 HE CPU, 3.60GHz, 4x16 ядер, 256 GB DDR3-1600.
- Производительность: 921,6 GFLOPS (14,4/ядро), 12 800 MB/s \* 4 канала \* 4 CPU (До 1600 MT/s, 140 GB/s согласно описанию платформы).
- Интерконнект: 4xBroadcom 5709C (Gigabit Ethernet).
- Операционная система: Ubuntu 18.04.1 LTS.
- Компилятор: gcc7.3.0-27.
- MPI: OpenRTE 3.0.0 (openmpi-based)

#### 3. Intel Endeavour (вычислительный кластер)

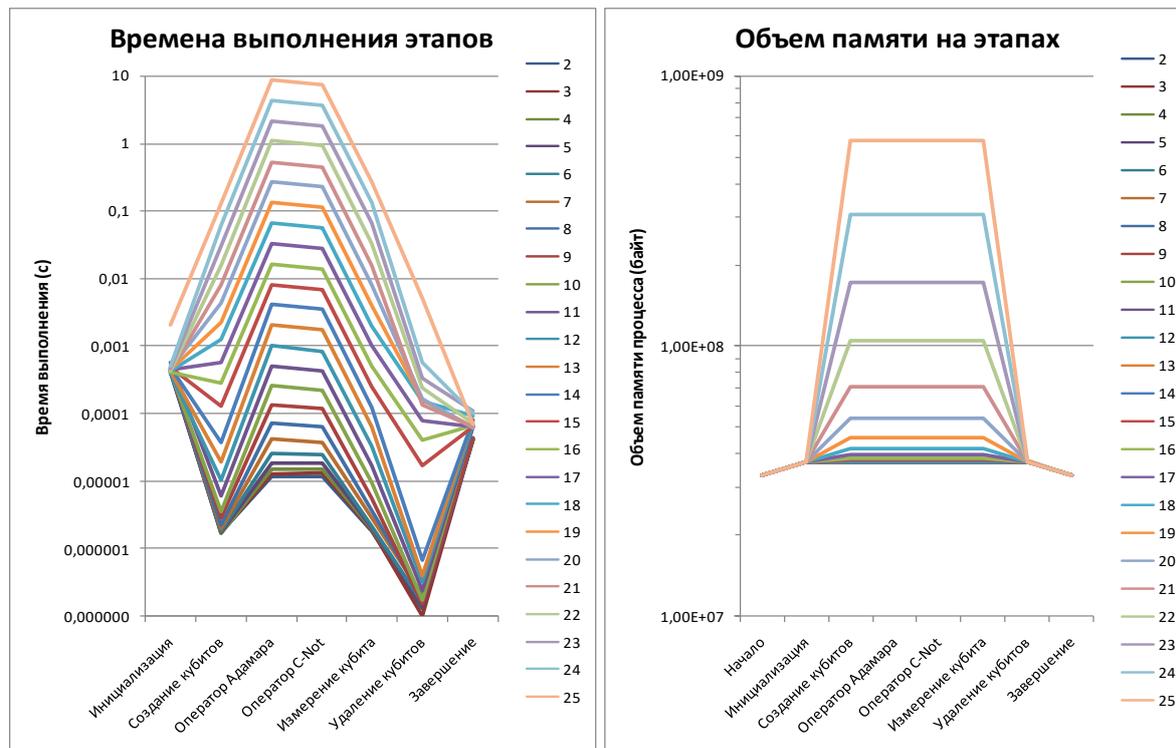
- Конфигурация: 2xIntel® Xeon® Platinum 8260L (Cascade Lake), 2.40GHz, 2x24 физических ядер, 256 GB DDR4-2933.
- Производительность: 2 304 GFLOPS (48/ядро), 23 466 MB/s \* 6 каналов \* 2 CPU.
- Интерконнект: Mellanox Technologies MT28908 Family [ConnectX-6] (HDR Infiniband).
- Операционная система: CentOS Linux 7 (Core).

- Компилятор: icc 19.0.5.281 20190815.
- MPI: Intel(R) MPI Library for Linux\* OS, Version 2019 Update 5 Build 20190806.

## 5. Результаты экспериментов

### 5.1 Виртуальная машина

В виртуальной машине проводился только первый эксперимент для квантовых систем размером от 2 до 25 кубитов. На рис.1 приведены результаты для пакета QuEST.



**Рис. 1.** Зависимость времени работы этапов алгоритма и объема используемой оперативной памяти от числа кубитов; пакет QuEST; виртуальная машина

Наблюдается экспоненциальная зависимость времени выполнения содержательных этапов от числа кубитов (отличия по времени для малого числа кубитов могут быть связаны с особенностями архитектуры системы, например, размещением всех данных в кеше). Объем потребляемой памяти приблизительно определяется как сумма константы (код и данные за исключением квантового состояния) и  $16 \cdot 2^N$  байт (размер вектора квантового состояния).

Времена выполнения операций инициализации и завершения не зависят от числа кубитов. Удаление кубита требует на 2-4 порядка меньше времени, чем остальные содержательные операции. Создание и измерение кубита требует в несколько десятков раз меньше времени, чем выполнение операторов. Оператор С-Not выполняется в 2-3 раза быстрее оператора Адамара.

Для других экспериментов и типов архитектур в большинстве случаев эти соотношения качественно сохраняются, поэтому далее мы их приводить не будем.

На рис.2 приведены результаты экспериментов для различных пакетов. Используется суммарное время выполнения всех этапов и максимальное потребление оперативной памяти.

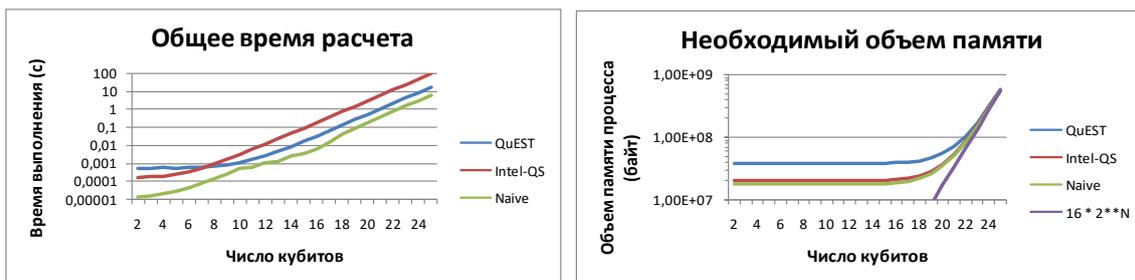


Рис. 2. Выполнение расчетов для различного числа кубитов; виртуальная машина

Экспоненциальная зависимость времени выполнения и объема потребляемой памяти от числа кубитов сохраняется, причем объем памяти фактически определяется размером вектора квантового состояния.

"Наивная" реализация показала более высокую производительность, чем другие пакеты. Таким образом, при моделировании простых квантовых схем собственные реализации могут показывать производительность, сравнимую с универсальными профессиональными пакетами.

## 5.2 Сервер Dell PowerEdge R815

На данном 64-ядерном сервере проводились эксперименты 1 и 2. На рис.3 приведены результаты первого эксперимента для квантовых систем размером от 2 до 33 кубитов.

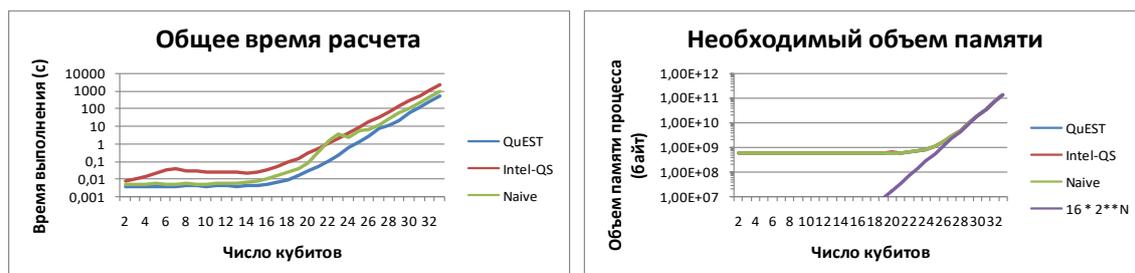


Рис. 3. Выполнение расчетов для различного числа кубитов; Dell PowerEdge R815

Скорость работы пакетов может отличаться в 4-5 раз. Низкая производительность Intel-QS по-видимому объясняется оптимизацией его кода для архитектур и компиляторов Intel.

Результаты эксперимента 2 по изучению эффективности сильной масштабируемости при распараллеливании на общей памяти приведены на рис.4.

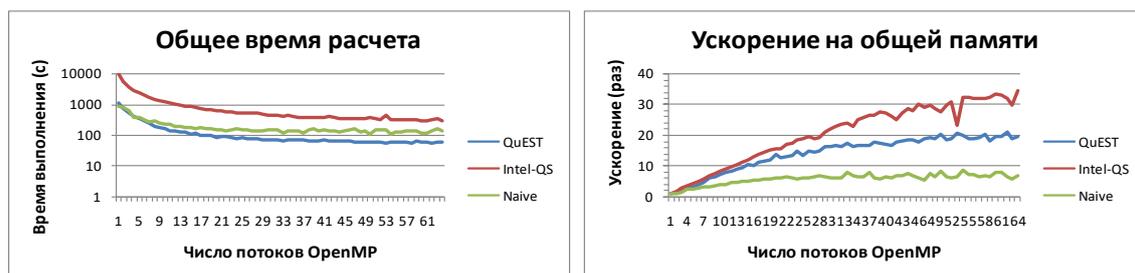


Рис. 4. Производительность пакетов на системах с общей памятью; Dell PowerEdge R815

Эффективность распараллеливания пакета QuEST, показавшего лучшие результаты на данной системе, составляет около 30%. Базовые квантовые гейты реализуются алгоритмами с низкой арифметической интенсивностью, поэтому их производительность ограничивается возможностями оперативной памяти, а не числом одновременно работающих потоков. Для более сложных схем продемонстрирована возможность достижения эффективности распараллеливания более 80% и производительности в 47% от теоретической [14].

## 5.2 Кластер Intel Endeavour

На вычислительном кластере Intel Endeavour проводились все эксперименты. На рис.5 приведены результаты первого эксперимента для квантовых систем размером от 2 до 33 кубитов.

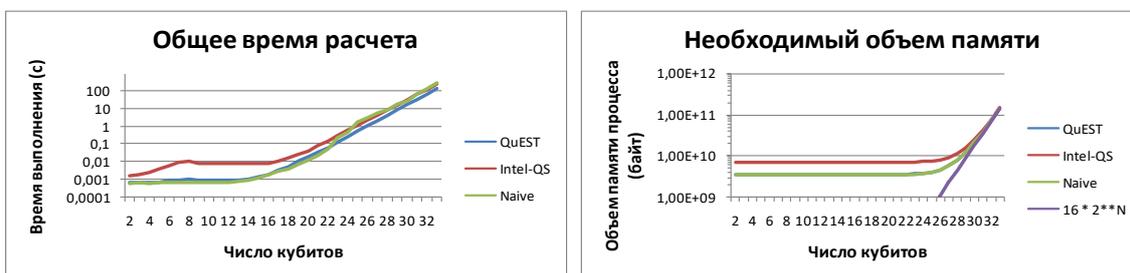


Рис. 5. Выполнение расчетов для различного числа кубитов; 1 узел Intel Endeavour

Для числа кубитов больше 20 скорость работы различных реализаций отличается не более чем в 2 раза.

На рис.6 показаны результаты эксперимента 2 по проверке эффективности сильной масштабируемости при распараллеливании на общей памяти.

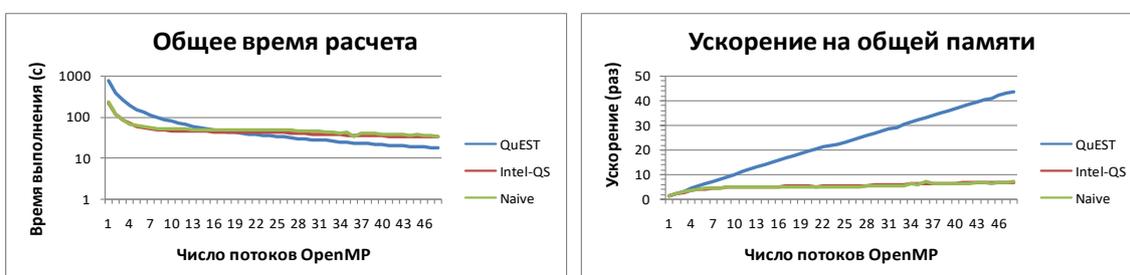


Рис. 6. Производительность пакетов на системах с общей памятью; 1 узел Intel Endeavour

QuEST показывает отличную эффективность распараллеливания (~90%), остальные пакеты – очень низкую (~15%). Для "наивной" реализации это объясняется отсутствием оптимизаций. Для Intel-QS – высокой производительностью однопоточной реализации и однопоточной реализацией операции вычисления вероятности нахождения кубита в одном из состояний. Без учета этой операции эффективность масштабирования Intel-QS составляет ~23%, а время выполнения при максимальном числе потоков больше чем у QuEST всего на 9%.

На рис.7 показаны результаты эксперимента 3 по изучению сильной масштабируемости при распараллеливании на распределенной памяти. Intel-QS не отработал при запуске с одним и двумя MPI-процессами. В целях оценки масштабируемости, для одного процесса приведены данные версии для общей памяти.

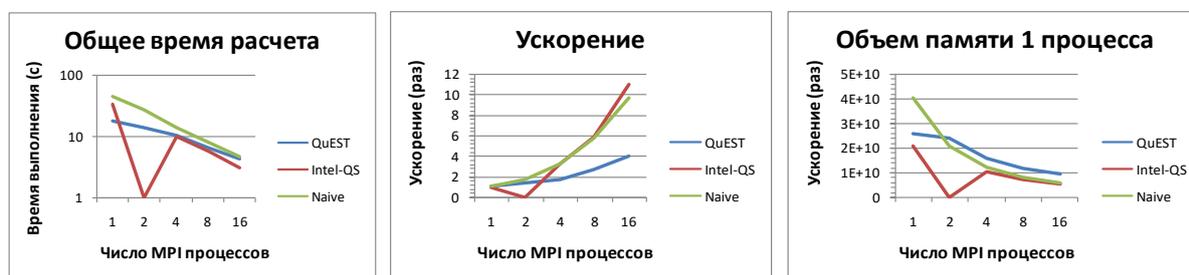


Рис. 7. Характеристики выполнения пакетов на системе с распределенной памятью; Intel Endeavour

Все реализации показывают сравнимую производительность, но QuEST показывает в 2,5 раза худшую масштабируемость. Время выполнения эксперимента слишком мало для точной оценки производительности, но при увеличении числа кубитов пакеты на доступных нам ресурсах перестают обрабатывать корректно для многих сочетаний параметров эксперимента.

Эффективность QuEST существенно отличается от заявляемой авторами; возможно, необходимо подбирать параметры его запуска или использовать более сложные квантовые схемы.

Объем памяти, используемой одним MPI-процессом, уменьшается примерно пропорционально их числу. Удвоение числа процессов позволяет сократить объем памяти каждого процесса примерно в два раза или использовать тот же объем для моделирования системы размером на один кубит больше.

## 6. Заключение

Результаты проведенных экспериментов позволяют сделать следующие выводы относительно возможностей симуляции работы квантовой схемы на идеальном квантовом компьютере с использованием различных вычислительных систем.

- Время моделирования экспоненциально зависит от числа кубитов.
  - Объем необходимой памяти экспоненциально зависит от числа кубитов и будет являться основным ограничивающим фактором при его увеличении.
  - Масштабируемость на общей памяти имеет среднюю эффективность (~30%), но известно, что на более сложных квантовых схемах можно достичь эффективности более 80%.
  - Эффективность масштабируемости на распределенной памяти достигает 70%.
  - Использование распределенной памяти позволяет увеличить размер моделируемых систем. При удвоении числа узлов можно симулировать систему размером на 1 кубит больше.
  - Переход от использования виртуальной машины с 1 ГБ оперативной памяти к суперкомпьютеру "Ломоносов-2" [24] позволит увеличить размер моделируемой системы с 25 до 40 кубитов, что не имеет принципиального значения при разработке, отладке и тестировании квантовых схем.
  - "Наивные" реализации могут быть использованы для моделирования простых квантовых схем и в целях обучения. Исследование больших и сложных схем может потребовать профессиональных пакетов.
  - В наших экспериментах Intel-QS и QuEST показали сравнимую производительность, лучший результат достигнут Intel-QS на распределенной памяти.
- Полученные результаты будут использованы для формирования набора инструментов, которые будут использоваться для разработки и анализа квантовых алгоритмов и обучения студентов и аспирантов ННГУ им. Н. И. Лобачевского.

## Литература

1. Mermin N. D. Quantum computer science: an introduction. – Cambridge University Press, 2007.
2. Trieu D. B. Large-scale simulations of error prone quantum computation devices. – Forschungszentrum Jülich, 2009. – Т. 2.
3. Wiki Q. List of QC simulators. – 2015. URL:<https://quantiki.org/wiki/list-qc-simulators> (дата обращения: 01.04.2020).
4. Green A. S. et al. Quipper: a scalable quantum programming language //Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation. – 2013. – С. 333-342.
5. Cross A. W. et al. Open quantum assembly language //arXiv preprint arXiv:1707.03429. – 2017.
6. Svore K. et al. Q# Enabling Scalable Quantum Computing and Development with a High-level DSL //Proceedings of the Real World Domain Specific Languages Workshop 2018. – 2018. – С. 1-10.
7. Abhari A. J. et al. Scaffold: Quantum programming language. – PRINCETON UNIV NJ DEPT OF COMPUTER SCIENCE, 2012.
8. Jones T. et al. Quest and high performance simulation of quantum computers //Scientific reports. – 2019. – Т. 9. – №. 1. – С. 1-11.

9. Guerreschi G. G. et al. Intel Quantum Simulator: A cloud-ready high-performance simulator of quantum circuits //Quantum Science and Technology. – 2020.
10. Smelyanskiy M., Sawaya N. P. D., Aspuru-Guzik A. qHiPSTER: the quantum high performance software testing environment //arXiv preprint arXiv:1601.07195. – 2016.
11. Aleksandrowicz G. et al. Qiskit: An open-source framework for quantum computing //Accessed on: Mar. – 2019. – Т. 16. URL: <https://zenodo.org/record/2562111> (Дата обращения: 01.04.2020).
12. Amy M., Gheorghiu V. staq--A full-stack quantum processing toolkit //arXiv preprint arXiv:1912.06070. – 2019.
13. Häner T., Steiger D. S. 5 petabyte simulation of a 45-qubit quantum circuit //Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. – 2017. – С. 1-10.
14. Pednault E. et al. Breaking the 49-qubit barrier in the simulation of quantum circuits //arXiv preprint arXiv:1710.05867. – 2017.
15. Pednault E., Gunnels J., Maslov D. and Gambetta J. On "Quantum Supremacy". – 2019. URL: <https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/>. (дата обращения: 01.04.2020).
16. Chen Z. Y. et al. 64-qubit quantum circuit simulation //Science Bulletin. – 2018. – Т. 63. – №. 15. – С. 964-971.
17. Arute F. et al. Quantum supremacy using a programmable superconducting processor //Nature. – 2019. – Т. 574. – №. 7779. – С. 505-510.
18. Google AI Blog: A Preview of Bristlecone, Google's New Quantum Processor. URL: <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html> (дата обращения: 01.04.2020).
19. The D-Wave 2000Q™ System. URL: <https://www.dwavesys.com/d-wave-two-system> (дата обращения: 01.04.2020).
20. de Avila A. B. et al. State-of-the-art quantum computing simulators: Features, optimizations, and improvements for D-GM //Neurocomputing. – 2019.
21. Häner T. et al. High performance emulation of quantum circuits //SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. – IEEE, 2016. – С. 866-874.
22. GitHub - QuEST-Kit/QuEST: A multithreaded, distributed, GPU-accelerated simulator of quantum computers. URL: <https://github.com/QuEST-Kit/QuEST> (дата обращения: 01.04.2020)
23. GitHub - iqusoft/intel-qs: High-performance simulator of quantum circuits. URL: <https://github.com/iqusoft/intel-qs> (дата обращения: 01.04.2020)
24. Vl. Voevodin, A. Antonov, D. Nikitenko, P. Shvets, S. Sobolev, I. Sidorov, K. Stefanov, Vad. Voevodin, S. Zhumatiy: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. In Journal: Supercomputing Frontiers and Innovations, Vol.6, No.2 (2019). pp.4–11.

# Статически-детерминированный метод прогнозирования динамических характеристик параллельных программ

А.А. Клейменов, Н.Н. Попова

Московский государственный университет имени М.В. Ломоносова

В статье рассматривается задача прогнозирования характеристик параллельных приложений. Изучаются динамические характеристики, описывающие выполнение параллельных приложений - время выполнения, количество операций с плавающей точкой, потребляемая электроэнергия, количество обращений в память и другие. Прогнозирование динамических характеристик позволяет решать многие проблемы, связанные с проектированием новых архитектур, выбором наиболее подходящих конфигураций многопроцессорных систем для решения конкретных задач, портированием приложений на новые системы, планированием потоков задач и многие другие. Задача прогнозирования характеристик активно исследуется. Возрастающая сложность архитектур современных высокопроизводительных систем требует разработки новых методов решения задачи прогнозирования. В статье дается обзор существующих подходов и программных средств для прогнозирования динамических характеристик и предлагается подход, основанный на статическом анализе исходного кода параллельного приложения. На основе текста параллельной программы, формального описания целевой вычислительной платформы и параметров запуска реализован метод, позволяющий прогнозировать время работы, количество выполненных операций вещественной арифметики, обращения к памяти и другие характеристики параллельного приложения. Применимость предложенного подхода продемонстрирована на примере решения тестовой трёхмерной задачи численного моделирования на многопроцессорном кластере на базе процессоров IBM Power8.

**Ключевые слова:** Параллельные приложения, динамические характеристики, анализ производительности, системы экзафлопсной производительности, модель компьютера, статический анализ

## 1 Введение

Одним из ключевых фактором, определяющим рост производительности вычислительных систем, является параллелизм обработки данных, поддерживаемый на всех уровнях организации вычислений. Именно благодаря параллелизму стало возможным преодоление барьеров, определяемых технологическими факторами. Современные суперкомпьютеры, производительность которых неуклонно приближается к экзафлопсам, включают в свой состав до миллиона ядер, обеспечивая тем самым высокую степень параллелизма.

Учитывая сложность суперкомпьютерных архитектур, трудно прогнозировать, насколько быстро приложение, предназначенное для современных суперкомпьютеров петафлопсной производительности, будет выполняться на суперкомпьютерах экзафлопсной производительности, каким будет потребление электроэнергии при этом и какие значения будут принимать другие параметры, характеризующие производительность параллельного приложения. Количественные характеристики, описывающие выполнение параллельных приложений, такие как, время выполнения, количество операций с плавающей точкой, потребляемая электроэнергия, объем использованной оперативной памяти и другие, будем называть *динамическими характеристиками*.

Точная экстраполяция характеристик параллельных программ, основанная на интуитивных представлениях и догадках, практически невозможна. Прогнозирование динамических характеристик параллельных программ является актуальной задачей. Прогнозирование необходимо для решения многих задач: разработки и оптимизации алгоритма для определённой архитектуры, оптимизации планирования задач, оптимизации размещения

компонентов приложения на гетерогенной архитектуре, выборе архитектуры, наилучшим образом подходящей для конкретного приложения и многих других.

Целью работы является разработка подхода к прогнозированию динамических характеристик параллельных программ, не требующего запуска программы на целевой вычислительной системе, простого в использовании и не требующего много времени для получения прогнозируемых характеристик. Во втором разделе статьи приводится классификация подходов к прогнозированию динамических характеристик параллельных программ. В третьем разделе описывается предлагаемый подход к анализу динамических характеристик параллельных программы и проводится его верификация.

## 2 Подходы к прогнозированию динамических характеристик

Задачу предсказания значения динамической характеристики  $h$  параллельной программы  $p$  при входных параметрах  $v \in V$  на вычислительной системе  $s$  можно представить как вычисление функции  $\hat{h} = \hat{H}(m, v)$ , где  $m$  – это модель, которая может быть представлена явно, как объединение двух моделей: модели программы  $m_p$  и модель вычислительной системы  $m_c$ .

Можно выделить три подхода к прогнозированию динамических характеристик (рис. 1): аналитический, симуляционный и гибридный. В аналитических подходах предсказание представляет собой вычисление аналитического выражения. В симуляционных подходах предсказание получается посредством симулирования программы или её редуцированного представления. Гибридные подходы используют как аналитические выражения, так и симуляцию для предсказания значения динамических характеристик.

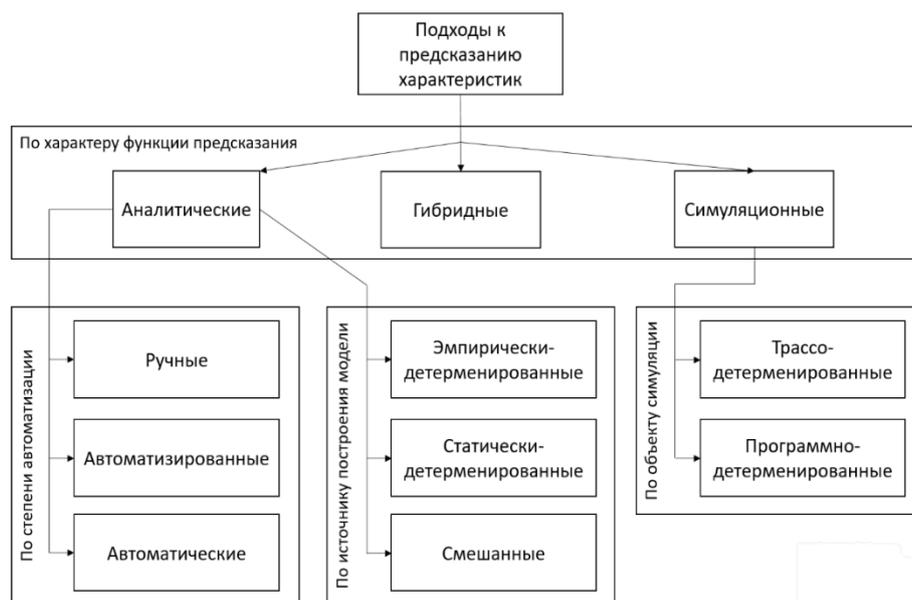


Рис. 1. Схема классификации подходов к предсказанию характеристик параллельных программ

### 2.1 Аналитические подходы

Основанием для построения модели прогнозирования могут являться как исходные тексты параллельных программ, так и информация о поведении программы, собранная во время ее выполнения. В зависимости от этого можно выделить три класса подходов к построению модели: статически-детерминированные (модель строится исходя только из исходного кода программы), эмпирически-детерминированные (модель определяется исходя из данных о результате выполнения программы на наборе параметров) и смешанные (используют как исходный код программы, так и данные о её выполнении).

В зависимости от степени автоматизации получения модели  $m$  аналитические подходы можно разделить на ручные, автоматизированные и автоматические.

Ручной подход предполагает, что исследователь создаёт модель вручную, исходя из своего понимания программы. Можно выделить наиболее общие шаги создания модели, характерные для этого подхода: выделение входных параметров, влияющих на значение исследуемой характеристики, выделение ядер, определение коммуникационного шаблона и возможности перекрытия выполнения передачи данных с вычислениями. У ручного подхода есть два основных недостатка: необходимость глубокого понимания работы программы исследователем и большая трудоёмкость.

Автоматизированные подходы более строго формализованы, что позволяет автоматизировать построение модели. С другой стороны, эти подходы зачастую требуют наличия исходного кода программы. В качестве примера автоматизированных подходов можно привести Modeling assertion framework (MA framework) [1] и Performance and architecture lab modeling tool (Palm) [2].

Автоматические подходы требуют минимального вмешательства со стороны исследователя, позволяя существенно облегчить разработку модели. Исследуемая программа в этом случае рассматривается как чёрный ящик. Автоматические подходы, в свою очередь, можно разделить на две группы: эмпирически-детерминированные и статически-детерминированные подходы. В эмпирически-детерминированных подходах модель строится исходя из данных, полученных при запуске параллельной программы. Статически-детерминированные подходы предполагают построение модели исходя из статического анализа исходного кода программы.

Эмпирически-детерминированные подходы хорошо изучены, что объясняется простотой их реализации. Наиболее часто встречающийся алгоритм построения модели в рамках данного подхода сводится к двум шагам: сбор значений исследуемой характеристики программы на подмножестве всевозможных значений входных параметров и обучение модели с помощью методов машинного обучения. В статье [3] проводится сравнение двенадцати различных алгоритмов машинного обучения на тестовом наборе мини-приложений Mantevo [4]. Исходя из результатов сравнения авторы делают вывод, что в случае плохо подобранных признаков сложные методы машинного обучения, например, глубокие нейронные сети требуют больший размер обучающей выборки и больше времени на обучение, но при этом имеют большую точность, чем такие простые методы, как МНК (метод наименьших квадратов).

Статически-детерминированные подходы сложны в своей реализации, но в отличие от эмпирически-детерминированных подходов они не требуют запуска целевой программы и зачастую не требуют наличия целевой вычислительной системы, что позволяет использовать их как инструмент суперкомпьютерного кодизайна. Для оценки сложных динамических характеристик рассматриваемые системы используют довольно простые модели, например, Roofline model [5]. В качестве немногочисленных представителей данной группы подходов можно назвать системы COMPASS [6] и ExaSAT [7]. COMPASS генерирует модель программы, используя расширение предметно-ориентированного языка для моделирования производительности Aspen [8], а ExaSAT использует специально разработанное представление модели в формате XML. Отметим, что свободный доступ к обеим упомянутым системам не предоставляется.

## 2.2 Симуляционные подходы

В зависимости от источника информации о симулируемых действиях данную группу подходов можно разделить на программно-детерминированные подходы и трассо-детерминированные подходы. Программно-детерминированные подходы симулируют программу или её редуцированное представление, а трассо-детерминированные подходы симулируют трассу событий. Трассо-детерминированные подходы зачастую работают быстрее. Они проще в реализации, чем подходы, симулирующие исполнение кода программы, но требуют получения исходной трассы путём исполнения программы или симуляции ее исполнения. Заметим, что трассы могут занимать много памяти. Существует множество

симуляторов, предназначенных для работы с последовательными программами. Однако, такие симуляторы непригодны для анализа параллельных приложений.

Примером симулятора, предназначенного для работы с приложениями для больших вычислительных систем, является симулятор BigSim [9]. Получение прогнозируемых характеристик с использованием BigSim можно описать следующим образом. Анализируемое параллельное приложение сначала эмулируется на малом количестве узлов. При этом обеспечивается сбор трассы приложения. Затем проводится симуляция полученной трассы. BigSim поддерживает до 100000 виртуальных MPI-процессов, распределённых по 2000 узлам.

Возможность рассматривать целевую программу в качестве чёрного ящика и отсутствие необходимости в наличии целевой машины являются главными преимуществами симуляционного подхода. Основным недостатком данной группы подходов является их времязатратность.

### 2.3 Гибридные подходы

Гибридные подходы пытаются уменьшить времязатратность симуляционных подходов и увеличить точность аналитических подходов, комбинируя элементы обоих подходов.

В качестве примера инструмента, реализующего гибридный подход, можно привести программный пакет PSINS [10], нацеленный на предсказание времени работы параллельных MPI-приложений. Модель приложения в рамках данного инструмента создаётся для каждого набора входных параметров. Модель включает в себя информацию о количестве обращений в память и их характере, количестве операций с плавающей точкой и трассы вызовов MPI-функций. Модель целевой вычислительной системы включает информацию о производительности (flop/s), пропускной способности памяти и характеристиках коммуникационной сети. Для получения времени работы приложения сначала определяется время выполнения последовательных участков трасс на целевой вычислительной системе. После этого трассы модифицируются в соответствии с полученными временами и симулируются.

## 3 Предлагаемый статически-детерминированный подход к прогнозированию динамических характеристик

### 3.1 Описание подхода

Разработка подхода для прогнозирования характеристик проводилась исходя из следующих требований: подход не должен требовать запуска программы на целевой машине, подход не должен быть трудоёмок для пользователя, предсказание характеристик не должно требовать много времени.

Исходя из представленных требований, предложенный подход можно классифицировать как статически-детерминированный. Существующие реализации статически-детерминированных подходов (COMPASS, ExaSAT) не имеют свободного доступа, что мотивировало разработку предлагаемого подхода.

Построение модели происходит на основе статического анализа исходного кода программы (рис. 2). Модель ВС и параметры запуска задаются пользователем, а затем подаются на вход инструментов предсказания динамических характеристик.

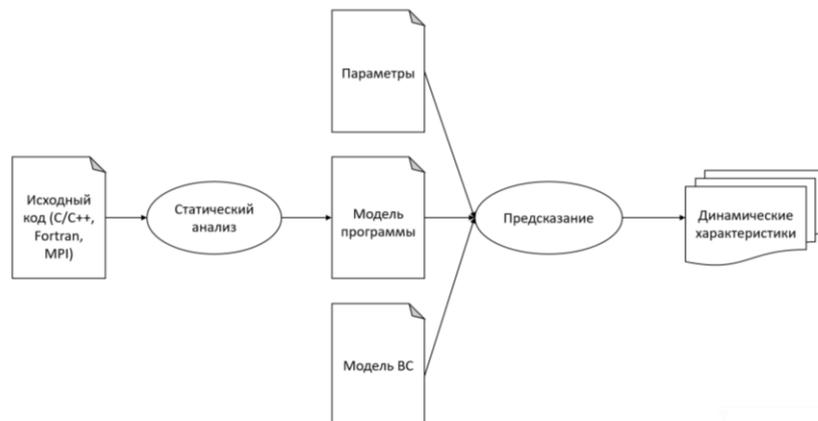


Рис. 2. Схема предлагаемого подхода

Модель программы описывается следующим образом:

$m_p = \langle f_1, f_2, \dots, f_n \rangle$ , где  $f_i$  - это модель  $i$ -ой функции программы. Функции упорядочены в соответствии с последовательностью их объявления в коде.

$f_i = \langle flops_i, loadBytes_i, storeBytes_i, mpiCalls_i, sentBytes_i, recvBytes_i, loadAccess_i, storeAccess_i, mathCalls_i \rangle$ , где

$flops_i = \langle flopsTotal_i, flopsSingle_i, flopsDouble_i \rangle$  – кортеж из функций, возвращающих общее количество операций с плавающей точкой, количество операций с числами одинарной точности и количество операций с числами двойной точности выполняемых функцией,

$loadBytes_i$  – количество считанных из памяти байт,  $storeBytes_i$  – количество записанных в память байт,

$mpiCalls_i = \langle send_i, recv_i, sendrecv_i, bcast_i, \dots \rangle$  – кортеж, содержащий информацию о количестве вызовов MPI-функций,

$sentBytes_i$  – количество отправленных по сети байт данных,  $recvBytes_i$  – количество принятых по сети байт данных,

$loadAccess_i$  – количество запросов на чтение из памяти,  $storeAccess_i$  – количество запросов на запись в память,

$mathCalls_i = \langle sin_i, cos_i, tan_i, \dots \rangle$  – кортеж, содержащий информацию о количестве вызовов математических функций.

На практике представленная модель программы генерируется статическим анализатором исходя из исходного кода программы. Модель представляется совокупностью python-классов, каждый из которых является моделью функции. Пример python-класса, представляющего фрагмент модели функции, приведен на рисунке 3.

```

class calculate:
    @staticmethod
    def flops(gv_NT, gv_1NX, gv_1NY, gv_1NZ, **kwargs):
        return 12 + (-2 + max(2, gv_NT)) * updateValues.flops(gv_1NX, gv_1NY, gv_1NZ)

    @staticmethod
    def flops4(**kwargs):
        return 0

    @staticmethod
    def flops8(gv_NT, gv_1NX, gv_1NY, gv_1NZ, **kwargs):
        return 12 + (-2 + max(2, gv_NT)) * updateValues.flops8(gv_1NX, gv_1NY, gv_1NZ)

    @staticmethod
    def load_access(gv_NT, gv_1NX, gv_1NY, gv_1NZ, **kwargs):
        return 28 + (-2 + max(2, gv_NT)) * (updateValues.load_access(gv_1NX, gv_1NY, gv_1NZ) +
        exchangeShadow.load_access(gv_1NX, gv_1NY, gv_1NZ) + 4)

    @staticmethod
    def store_access(gv_NT, gv_1NX, gv_1NY, gv_1NZ, **kwargs):
        return 24 + (-2 + max(2, gv_NT)) * (updateValues.store_access(gv_1NX, gv_1NY, gv_1NZ) +
        exchangeShadow.store_access(gv_1NX, gv_1NY, gv_1NZ) + 2)
  
```

Рис. 3. Пример фрагмента модели функции

Модель ВС описывается следующим образом:

$m_c = \langle netLatency, netBandwidth, n_1, \dots, n_k \rangle$ , где

$netLatency$  – латентность сети,  $netBandwidth$  – пропускная способность сети,  $n_1, \dots, n_k$  – модели узлов вычислительной системы.

$n_i = \langle coreCount, performance, memBandwidth, mathFlops \rangle$ , где  $coreCount$  – кол-во ядер на узле,  $performance$  – производительность узла (в Gflop/s),  $memBandwidth = \langle l1Bandwidth, l2Bandwidth, l3Bandwidth, dramBandwidth \rangle$  – пропускные способности различных уровней памяти,  $mathFlops$  – кортеж с оценками количества операций с плавающей точкой, выполняемых математическими функциями.

Модель ВС также реализуется как python класс. Пример модели вычислительной системы, состоящей из пяти 20-ядерных узлов, приведён на рисунке 4.

```
class Polus:
    @staticmethod
    def isMultiNode():
        return True

    @staticmethod
    def nodeCount():
        return 4

    @staticmethod
    def getNode(index):
        return PolusNode

    @staticmethod
    def networkLatency():
        return 60

    @staticmethod
    def networkBandwidth():
        return 10000

class PolusNode:
    @staticmethod
    def isMultiNode():
        return False

    @staticmethod
    def coreCount():
        return 20

    @staticmethod
    def l1Bandwidth():
        return 13000

    @staticmethod
    def dramBandwidth():
        return 9000

    @staticmethod
    def performance():
        return 5.3

    @staticmethod
    def flopsPerSin8():
        return 39

    @staticmethod
    def flopsPerCos8():
        return 39

    @staticmethod
    def flopsPerTan8():
        return 39

    @staticmethod
    def flopsPerSqrt8():
        return 1

    @staticmethod
    def flopsPerLog8():
        return 39

    @staticmethod
    def flopsPerLog10_8():
        return 39
```

Рис. 4. Пример модели вычислительной системы

Прогнозирование времени работы MPI-приложения, исходя из модели программы и модели вычислительной системы, проводится следующим образом:

$T_{mpi} = \max(T_1, T_2, \dots, T_n)$ , где  $T_i$  – время работы  $i$ -го MPI-процесса.

$T_i = T_{comm_i} + T_{comp_i}$ , где  $T_{comp_i}$  – время, затрачиваемое на вычисления,  $T_{comm_i}$  – время, затрачиваемое на межсетевые коммуникации.

$T_{comp_i} = \max\left(\frac{flops_i}{perf_i}, \frac{mem_i}{bandwidth_i}\right)$ , где  $flops_i$  – количество операций с плавающей точкой, выполненных  $i$ -ым MPI процессом,  $perf_i$  – производительность ядра, на котором выполняется  $i$ -ый MPI процесс,  $mem_i$  – объём записанных и прочитанных данных,  $bandwidth_i$  – пропускная способность памяти. Подставляя пропускные способности различных уровней памяти, получаем времена работы при условии, что все запросы в память попадают в этот уровень памяти.

$T_{comm_i} = callCount_i * netLatency + \frac{netSize_i}{netBandwidth}$ , где  $callCount_i$  – количество вызовов MPI функций,  $netLatency$  – латентность сети,  $netSize_i$  – размер передаваемых и получаемых данных,  $netBandwidth$  – пропускная способность сети. Отметим, что такой способ оценки времени коммуникационного взаимодействия не очень точен. Однако, он требует минимум информации о вычислительной системе. В отличие от симуляции он является простым и быстрым в применении.

### 3.2 Верификация подхода

Верификация предложенного подхода проводилась на вычислительном кластере Polus[11]. Исследовались динамические характеристики программы, моделирующей распространение волны в трёхмерном пространстве. Вычислительный кластер Polus состоит из пяти узлов, один из которых является головным. На каждом узле установлено два процессора IBM Power8 с графическими процессорами NVIDIA Tesla P100. Для получения пропускных способностей уровней памяти и производительности ядер использовался Empirical Roofline Tool [12]. Для

определения латентности и пропускной способности сети использовался тестовый пакет OSU Micro-Benchmarks [13].

Программа, моделирующая распространение волны в трёхмерном пространстве, основана на методе конечных разностей с использованием регулярной трёхмерной сетки. Сетка разбивается на трёхмерные блоки и равномерно распределяется по MPI-процессам. В каждой точке сетки итерационно рассчитываются значения искомым функций согласно заданному семиточечному разностному оператору. Для расчета значения переменной в каждой точке сетки требуется значения шести переменных в соседних точках сетки. На каждой итерации по времени процессы рассчитывают свою часть сетки и обмениваются боковыми гранями с шестью соседними процессами.

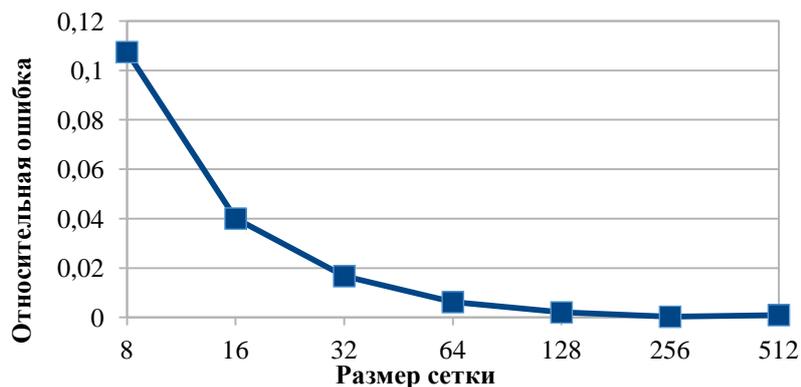


Рис. 5. Относительная ошибка предсказания количества операций с плавающей точкой.

Запуски программы проводились на кубических сетках с размерами: 8x8x8, 16x16x16, 32x32x32, 64x64x64, 128x128x128, 256x256x256, 512x512x512. Для сбора информации о количестве операций с плавающей точкой отслеживалось аппаратное событие PM\_FLOP. На рис.5 представлена относительная ошибка предсказания количества операций с плавающей точкой, рассчитываемая как  $E_{rel} = |(h - \hat{h})/h|$ , где  $h$  – реальное значение характеристики,  $\hat{h}$  – предсказанное значение характеристики. Из рисунка видно, что максимальное значение относительной ошибки достигается при минимальных размерах сетки и значение ошибки существенно уменьшается с увеличением размера сетки. Этот эффект объясняется тем, что доля математических функций (sin, cos), используемых для инициализации сетки, больше на маленьких сетках. Количество же операций с плавающей точкой, выполняемых этими функциями, зависит от вычисляемого значения, из-за чего погрешность предсказания для таких функций больше.

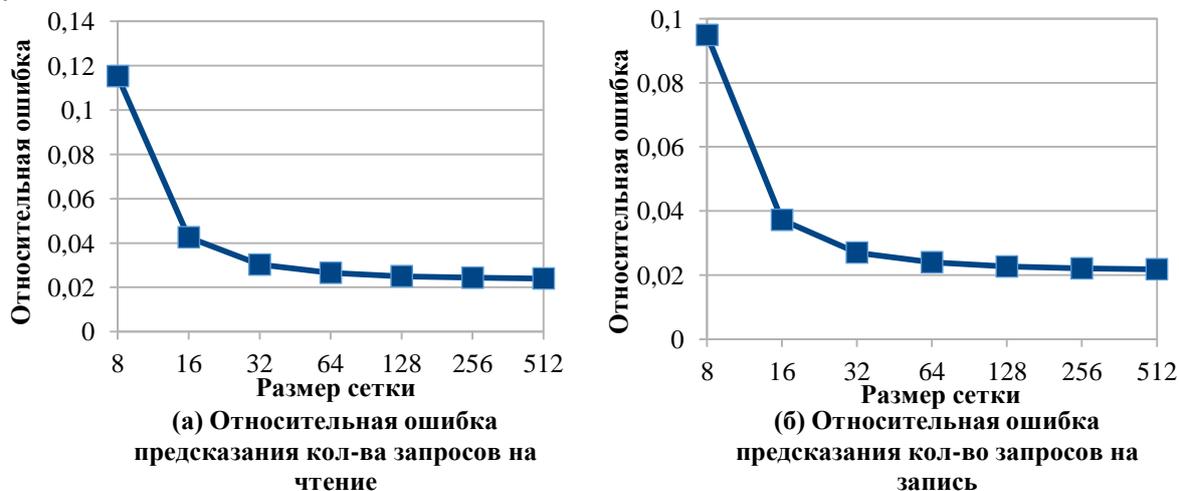
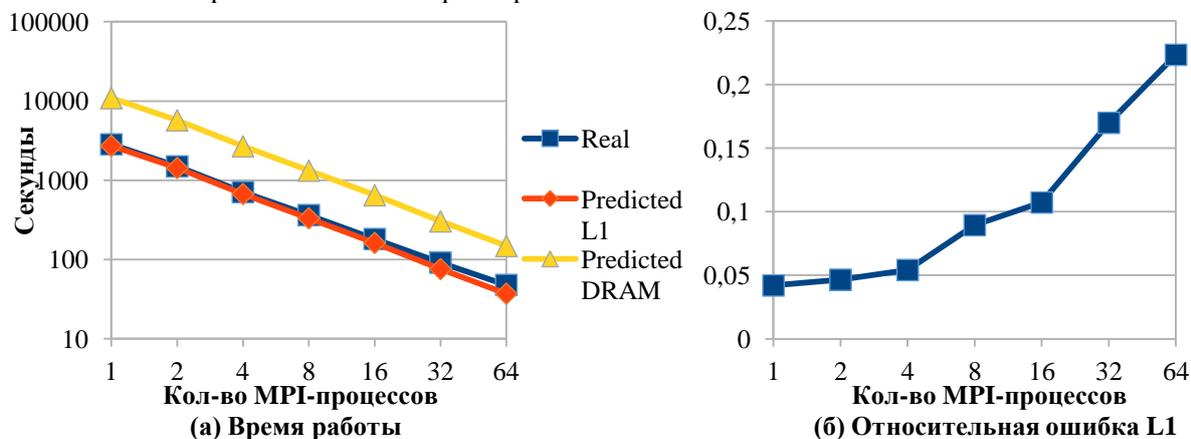


Рис. 6. Относительная ошибка предсказания количества запросов на чтение (а), относительная ошибка предсказания количества запросов на запись (б).

На рисунке 6 представлены графики относительной ошибки предсказания количества операций доступа к памяти на чтение и запись. Для измерения количества запросов на чтение и запись использовались аппаратные события PM\_LD\_CMPL и PM\_ST\_CMPL соответственно. Погрешность при маленьких размерах сетки объясняется большей долей математических функций, а также обращениями к памяти, совершаемыми в процессе динамической линковки, количество которых не зависит от размера сетки.



**Рис. 7.** Измеренное и предсказанное время работы при условии, что все обращения в память попадают в кэш L1 или оперативную память (а), относительная ошибка предсказания времени работы при условии, что все обращения попадают в L1 кэш (б).

Предсказание времени работы проводилось для сетки размером 512x512x512. На рисунке 7(а) представлены графики времени выполнения реального приложения, а также предсказанного времени выполнения при условии, что все обращения в память попадают в кэш L1 или в оперативную память. На графике 7(б) представлена относительная ошибка предсказания времени выполнения при условии, что все обращения попадают в кэш L1. Из рисунка видно, что реальное время выполнения очень близко к предсказанному времени работы при попадании всех обращений в кэш L1. Исходя из этого, можно сделать вывод о том, что программа обладает хорошей локальностью обращений в память. Рост ошибки предсказания времени работы с увеличением количества процессов можно объяснить ростом погрешности в предсказании времени коммуникаций.

## 4 Заключение

В статье представлено описание разрабатываемого подхода для прогнозирования динамических характеристик параллельных программ, основанного на статическом анализе исходного кода. Предлагаемый подход позволяет предсказывать значение таких характеристик параллельных программ, как число обращений на чтение и запись в память, число считанных и записанных в память байт, количество операций с плавающей точкой, размер отправленных и принятых по сети данных, а также время работы программы. Верификация предложенного подхода продемонстрирована на параллельной программе, моделирующей распространение волны в трёхмерном пространстве. Относительная ошибка прогнозирования динамических характеристик составила менее 25%.

Дальнейшее направление исследований предполагает верификацию приведённого подхода на ВС с процессорами Intel и AMD, расширение подхода для предсказания энергопотребления, повышения точности предсказания коммуникационных расходов и работу с приложениями, использующими графические ускорители.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 20-07-01053).

## ЛИТЕРАТУРА

1. Alam S.R., Vetter J.S. A framework to develop symbolic performance models of parallel applications // Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. IEEE, 2006. Vol. 2006. P. 1–8. DOI:10.1109/IPDPS.2006.1639625.
2. Tallent N.R., Hoisie A. Palm: Easing the Burden of Analytical Performance Modeling // Proc. 28th ACM Int. Conf. Supercomput. - ICS '14. 2014. P. 221–230. DOI:10.1145/2597652.2597683.
3. Malakar P. et al. Benchmarking Machine Learning Methods for Performance Modeling of Scientific Applications // 2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS). IEEE, 2018. P. 33–44. DOI:10.1109/PMBS.2018.8641686.
4. Mantevo Project [Электронный ресурс]. URL: <https://mantevo.github.io> (дата обращения: 03.03.2020).
5. Williams S., Waterman A., Patterson D. Roofline: an insightful visual performance model for multicore architectures // Commun. ACM. 2009. Vol. 52, № 4. P. 65–76. DOI:10.1145/1498765.1498785.
6. Lee S., Meredith J.S., Vetter J.S. COMPASS: A Framework for Automated Performance Modeling and Prediction // Proceedings of the 29th ACM on International Conference on Supercomputing - ICS '15. New York, New York, USA: ACM Press, 2015. P. 405–414. DOI:10.1145/2751205.2751220.
7. Unat D. et al. ExaSAT: An exascale co-design tool for performance modeling // Int. J. High Perform. Comput. Appl. 2015. Vol. 29, № 2. P. 209–232. DOI:10.1177/1094342014568690.
8. Spafford K.L., Vetter J.S. Aspen: A domain specific language for performance modeling // 2012 International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2012. P. 1–11. DOI:10.1109/SC.2012.20.
9. Gengbin Zheng, Kakulapati G., Kale L.V. BigSim: a parallel simulator for performance prediction of extremely large parallel machines // 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings. IEEE, 2004. P. 78–87. DOI:10.1109/IPDPS.2004.1303013.
10. Tikir M.M. et al. PSINS: An open source event tracer and execution simulator // Dep. Def. Proc. High Perform. Comput. Mod. Progr. - Users Gr. Conf. HPCMP-UGC 2009. 2009. P. 444–449. DOI:10.1109/HPCMP-UGC.2009.73.
11. Polus [Электронный ресурс]. URL: <http://hpc.cmc.msu.ru/polus> (дата обращения: 03.03.2020).
12. Empirical Roofline Tool [Электронный ресурс]. URL: <https://crd.lbl.gov/departments/computer-science/PAR/research/roofline/software/ert/> (дата обращения: 03.03.2020).
13. OSU Micro-Benchmarks [Электронный ресурс]. URL: <http://mvapich.cse.ohio-state.edu/benchmarks/> (дата обращения: 03.03.2020).



**Суперкомпьютерные дни в России**

**Аннотации стендовых  
докладов**

# Improving performance of linear solvers by using indices compression for storing sparse matrices\*

S. Zarechnev, B. Krasnopolsky

Institute of Mechanics, Lomonosov Moscow State University

The demands for solving systems of linear algebraic equations (SLAEs) appear in a whole lot of mathematical modeling problems. The choice of the optimal method and corresponding implementation typically depends on the nature of the SLAE system and the type of the differential equation producing this matrix. A couple of highly scalable open-source libraries are known to date (e.g., **hypre**, **PETSc**, or **Trilinos**) and are widely used in the simulation codes. However, there are still cases when development of novel libraries can be a reasonable practice.

The **XAMG** library is an example of such a newly developing library of numerical methods for solving systems of linear algebraic equations. The library is intended for solving series of SLAEs with constant matrix and multiple right-hand side (RHS) vectors, which occur, for example, in incompressible turbulent flow simulations [1]. It contains implementation of a set of Krylov subspace iterative methods (including the merged formulations [2]) and multigrid methods, and provides pseudo-block variants of the methods to solve systems with multiple RHS vectors. The **hypre** library is used to construct the multigrid matrix hierarchy during the setup phase, and the **XAMG** focuses on optimized implementation for the solution phase of the corresponding methods. The multilevel parallelization approach [3] with three logical levels, corresponding to compute node / NUMA node / CPU core, allowing to minimize the amount of inter-node MPI communications, is applied. The intra-node communications and synchronizations are implemented with help of POSIX shared memory and atomic operations. The current implementation outperforms the **hypre** library by at least of 20-30% and still has a great potential for further optimization.

The present work is aimed to consider one of such optimizations related to compression of indices, used to store the sparse matrices. The list of basic operations used in Krylov subspace and multigrid methods includes linear operations with vectors, dot products and matrix-vector multiplications. All of them provide very low FLOP per byte ratio and are memory bound [4]. The matrix slice corresponding to each compute process finally is segmented on up to several tens of sub-blocks, stored in the CSR format. Using compression of empty rows and columns, one can obtain the blocks with number of rows, columns, and nonzero elements, which can be expressed by 2-byte or even 1-byte integer numbers.

Simple theoretical considerations allow to estimate the expected speedup due to the suggested optimization, dealing with reduction of integer types size, by about 10-15%, which can be an attractive and easily achievable goal. The corresponding estimates are confirmed by the numerical simulation results with separate test code. Results of the numerical experiments validating the suggested optimization for **XAMG** library, which is expected to be of the same order of 10-15%, will be presented in the conference poster.

## References

1. Krasnopolsky B. An Approach for Accelerating Incompressible Turbulent Flow Simulations Based on Simultaneous Modelling of Multiple Ensembles // Computer Physics Communications. 2018. Vol. 229. P. 8-19.
2. Krasnopolsky B. Revisiting performance of BiCGStab methods for solving systems with multiple right-hand sides // Computers & Mathematics with Applications. 2020.

---

\*The presented work is supported by the RSF grant No. 18-71-10075.

Vol. 79 (9). P. 2574–2597.

3. Krasnopolsky B., Medvedev A. Acceleration of Large Scale OpenFOAM Simulations on Distributed Systems with Multicore CPUs and GPUs // *Advances in Parallel Computing. Parallel Computing: On the Road to Exascale*. 2016. Vol. 27. P. 93–102.
4. Williams S., Waterman A., Patterson D. Roofline: An insightful visual performance model for multicore architectures // *Communications of the ACM*. 2009. Vol. 52 (4) P. 65–76.

## Высокопроизводительная технология сопоставления изображений на основе оптического потока и метода согласованных оценок\*

К.Г. Пугачев<sup>1</sup>, А.П. Котов<sup>1,2</sup>, В.А. Фурсов<sup>1,2</sup>

Самарский национальный исследовательский университет  
имени академика С.П. Королева<sup>1</sup>, Институт систем обработки изображений РАН<sup>2</sup>

Задача сопоставления изображений с целью нахождения соответствующих точек на различных изображениях одной сцены является одной из центральных проблем при решении ряда практических задач: построение трехмерных моделей сцен, «сшивка» изображений для получения панорамных изображений и ортофотопланов, определение движения объектов, поиск ориентиров в системах автономной навигации и др. Актуальность этой задачи возрастает в связи с широким внедрением систем технического зрения во все сферы жизни.

Мы предлагаем новую высокопроизводительную технологию сопоставления изображений, основанную на последовательном использовании оптического потока и метода согласованных оценок. Технология предназначена для определения относительных сдвигов на двух изображениях, которые для краткости изложения далее названы *первое* и *второе* соответственно. Предполагается, что яркость постоянна, т.е. пиксель переходит в следующий кадр без изменений:

$$I(x, y, t) \cong I(x + \Delta x, y + \Delta y, t + \Delta t).$$

На первом этапе решается задача определения относительных сдвигов с использованием алгоритма, построенного на основе алгоритма оптического потока Лукаса–Канаде [1]. Далее с использованием полученной карты сдвигов и исходных изображений формируется так называемое *разностное* изображение.

Если оценки сдвигов в результате применения метода оптического потока совпадают с истинными значениями относительных сдвигов, разностное изображение будет заполнено нулевыми значениями. В действительности этого не происходит вследствие неизбежных методических и вычислительных ошибок.

Для реализации этапа уточнения соответствующих точек мы формируем бинарное изображение, на котором яркие пиксели соответствуют областям, в которых значение яркости разностного изображения превышает допустимое значение.

Далее мы осуществляем поиск соответствующей точки в малой окрестности точки, сдвинутой на величину, найденную на первом этапе. В данном случае мы предполагаем: хотя в данной точке имеет место ошибка в определении сдвига, тем не менее эта точка все равно ближе к истинной, т.е. имеет место ошибка отклонения.

Для уточнения сдвигов в областях ошибок мы применяем метод согласованных оценок, предложенный в работе [2]. Суть метода заключается в следующем. Пусть  $\mathbf{f}_i^1$  и  $\mathbf{f}_j^2$  – векторы дескрипторов, полученных, трансформацией в вектор фрагментов, описывающих окрестности рассматриваемых точек  $i$  и  $j$ , принадлежащих первому и второму изображениям соответственно. Если каждый дескриптор представляет собой вектор из  $K$  элементов, то согласованная мера близости определяется как:

$$W_{i,j} = \sum_{\substack{s=1 \\ p=s+1}}^K (\Delta f_s^{i,j} - \Delta f_p^{i,j})^2, \quad (1)$$

где  $\Delta f_s^{i,j}$ ,  $\Delta f_p^{i,j}$  –  $s$ -й и  $p$ -й элементы вектора разностей  $\Delta \mathbf{f}^{i,j} = \mathbf{f}_i^1 - \mathbf{f}_j^2$ .

Значения функции согласованности (1) вычисляются для всех точек из области поиска, сформированной в окрестности точки на втором изображении, смещенной на величину сдвига, полученного на первом этапе (определения оптического потока). Точка  $j$  второго изображения

\* Работа выполнена при финансовой поддержке РФФИ (проект №17-29-03112) и Министерства науки и высшего образования (госзадание).

выбирается соответствующей для точки  $i$  первого изображения, если среди всех точек  $j'$  в заданной области поиска  $D$  на втором изображении её значение показателя согласованности  $W_{i,j}$  оказалось минимальным.

Вычисление оптического потока и уточнение сдвигов с помощью метода согласованных оценок являются довольно затратными по времени операциями. Для повышения быстродействия сопоставления изображений была разработана параллельная программа, реализованная на языке программирования C++ с применением технологии OpenMP. В соответствии с описанной технологией этапы определения оптического потока, формирования разностного бинарного шаблона и уточнения сдвигов реализуются последовательно. Для ускорения первого этапа использовался параллельный алгоритм решения переопределенных систем уравнений на каждом анализируемом фрагменте. На этапе уточнения относительных сдвигов с использованием метода согласованных оценок в предложенном параллельном алгоритме вычисление значения сдвига каждого пикселя выполняется параллельно и не требует пересылок данных между потоками. Общая блок-схема алгоритма представлена на рисунке 1. Программа была запущена на ноутбуке под управлением операционной системы Windows 10 с четырехъядерным процессором Intel Core i7-3632QM с тактовой частотой 2,2 ГГц с восемью логическими ядрами, а также с 8 Гб оперативной памяти. Полученные графики ускорения и эффективности [3] приведены на рисунке 2.



Рис. 1. Блок-схема алгоритма

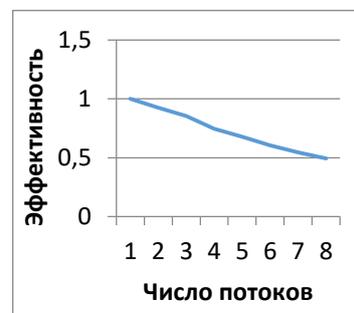
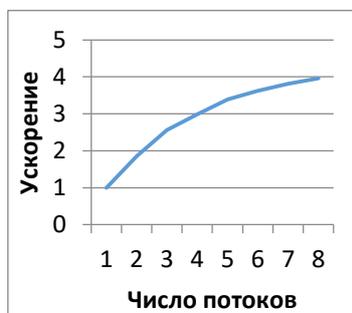


Рис. 2. Ускорение и эффективность

Предложенная технология является эффективной во многих отношениях.

1. Применение на первом этапе алгоритма оптического потока позволяет строить плотное множество соответственных точек на участках фона.

2. На этапе уточнения сдвигов в точках, соответствующих областям ошибок (ярких точках разностного шаблона), позволяет существенно сократить вычислительные затраты, т.к. значительная часть точек исключается из обработки.

Несмотря на сокращение вычислительных затрат удается достичь высокой точности сопоставления, т.к. на этапе уточнения используется эффективный метод согласованных оценок, который обеспечивает высокую точность определения больших сдвигов, при которых оптический поток неработоспособен. Предложенная высокопроизводительная технология может представлять интерес для сопоставления изображений в мобильных системах реального времени.

## Литература

1. Dhara P., Saurabh U. Optical Flow Measurement using Lucas Kanade Method // International Journal of Computer Applications. 2013. Vol. 61 No. 10. P. 6-10. DOI: 10.5120/9962-4611.
2. Fursov V.A., Goshin Ye.V., Pugachev K.G. Adaptive algorithm of conforming image matching // Proceedings of the Data Science Session at the V International Conference on Information Technology and Nanotechnology, ITNT 2019, May 21-24, 2019, Samara, Russia. SEUR, 2019. P. 26-33. DOI: 10.18287/1613-0073-2019-2416-26-33.
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с. ISBN 5-94157-160-7.

## Использование MPI- и GPU-ускоренного комбинированного метода квантовой механики, молекулярной механики и метадинамики для моделирования стадии окисления субстрата в каталитическом цикле 2-гидроксибифенил-3-монооксигеназы из *Pseudomonas azelaica*\*

К.Е. Копылов, Е.М. Кирилин, В.К. Швядас

Факультет биоинженерии и биоинформатики и НИИ ФХБ имени А.Н. Белозерского МГУ имени М.В. Ломоносова

С использованием суперкомпьютерных технологий проведено моделирование окисления субстрата с помощью активного интермедиата FAD-C4a-гидропероксида – ключевой стадии каталитического механизма действия фермента 2-гидроксибифенил-3-монооксигеназы (HbrA) из почвенной бактерии *Pseudomonas azelaica* [1]. HbrA представляет интерес как перспективный биологический катализатор в фармацевтической промышленности для получения различных фенольных соединений [1]. Производство фермента и его эксплуатация в технологических процессах не представляют существенных затруднений, однако для широкого практического использования необходимо расширить его субстратную специфичность. Природным субстратом фермента является 2-гидроксибифенил; с помощью эмпирических подходов получен ряд мутантов, обладающих активностью к некоторым другим субстратам [1,2], однако такой профиль превращаемых субстратов недостаточен, а дальнейшее целенаправленное изменение свойств фермента сдерживает отсутствие понимания каталитического механизма действия на молекулярном уровне.

Для решения задачи использовали комплексный подход, заключающийся в применении интеграции современных программных пакетов квантово-механического моделирования (CP2K [3], Gaussian), молекулярной динамики (Amber) [4] и метадинамики (Plumed) [5]. В качестве метода моделирования был выбран комбинированный метод квантовой и молекулярной механики (QM/MM) и метадинамики: расчёты выполнены как с помощью полумпирических методов PM6-DH+ и DFTB3, так и метода функционала плотности PBE в базе 6-311++G\*\*. Квантомеханическая часть состояла из 85 атомов: изоаллоксазинового кольца FAD-C4a-гидропероксида, природного субстрата 2-гидроксибифенила и каталитических остатков активного центра His48, Asp117 и Asp222. В качестве координат реакции (коллективных переменных) были выбраны: (а) для полуреакции гидроксирования – разность длин разрываемой связи O—O в пероксидной группе FAD и образуемой связи C—O в субстрате и (б) для полуреакции депротонирования – разность длин разрываемой связи O—H в субстрате и образуемой связи N—H с  $\epsilon$ -азотом каталитического остатка His48. Общий размер системы составляет 85 тысяч атомов.

Полуэмпирическое моделирование осуществляли с помощью программ *sander* из пакета AmberTools 19/Amber 18 [4] и CP2K 7.0 [3], модифицированных движком метадинамики Plumed 2 [5], собранным с использованием OpenMPI. Для увеличения степени исследования системой фазового пространства вычисления запускали в 5 параллельных расчётах с обобщённым дополнительным потенциалом («walkers» по терминологии Plumed), представленных отдельными задачами в системе очередей SLURM на суперкомпьютере «Ломоносов-2».

Конфигурации сборки и запуска выбирали следующим образом:

- пакет AmberTools 19/Amber 18 собирали компиляторами GCC (gcc, g++, gfortran) с поддержкой OpenMPI, библиотеки Intel MKL и движка метадинамики Plumed 2.4, собранного с поддержкой OpenMPI. Для моделирования использовали MPI-версию программы *sander*. Графические процессоры не использовались в связи с отсутствием реализации

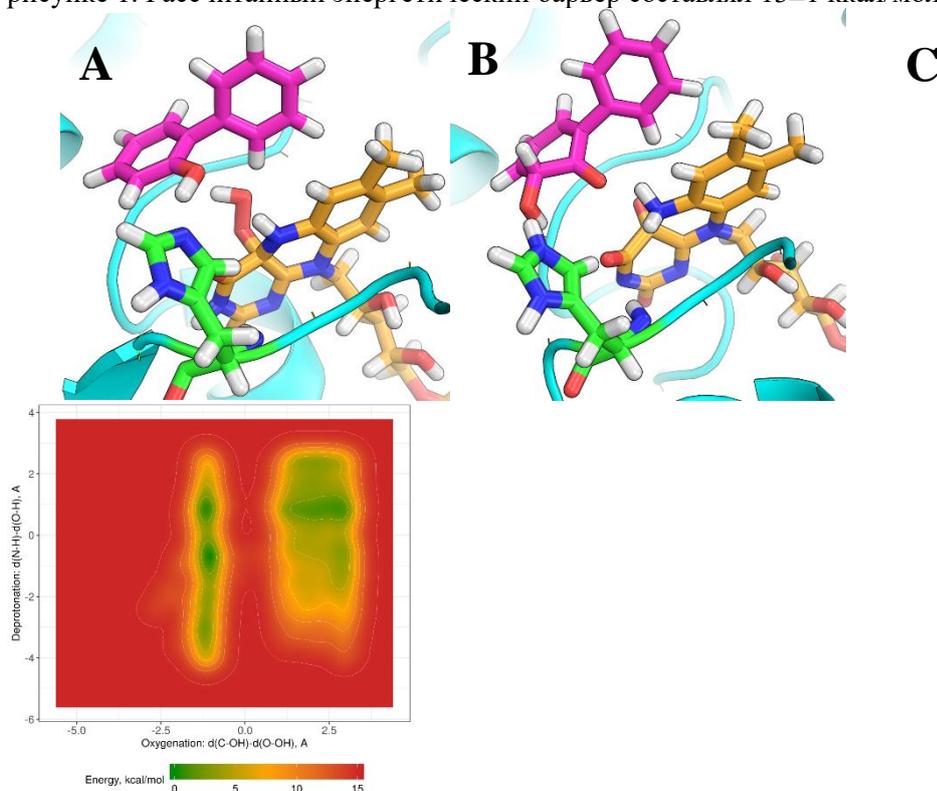
---

\* Работа выполнена при финансовой поддержке РФФИ (грант №17-07-00751 А)

квантовомеханических алгоритмов в GPU-ускоренной версии. Варианты, собранные компиляторами Intel версий 14-16, показали нестабильную работу при задействовании встроенного движка полуэмпирической квантовой механики. На одну задачу-«walker» выделялось 8 узлов очереди «compute» (каждый узел содержит один процессор Intel Xeon E5-2697 v3, 64 ГБ оперативной памяти и один графический сопроцессор NVIDIA Tesla K40). На каждом узле запускалось по 14 однопоточных MPI-процессов.

• Пакет CP2K 7.0 также собирали компиляторами GCC, с поддержкой CUDA 8.0, OpenMPI, Intel MKL и Plumed 2.4. При сборке были задействованы параллельные вычислительные библиотеки: ScaLAPACK из пакета Intel MKL, ELPA, SuperLU, PT-SCOTCH, PARMETIS. В расчётах использовался модуль *cp2k.psmr*. Для полуэмпирических расчётов на одну задачу-«walker» выделялось по 8 узлов очереди «compute», для метода функционала плотности (PBE) – по 16; на каждом узле запускалось 14 MPI-процессов по 2 SMP-потока.

В задаче полуэмпирического моделирования пакет Amber показал существенно лучшую сходимость и более высокую скорость счёта, несмотря на отсутствие поддержки графических процессоров: 50 пс/сутки для пакета Amber против 2 пс/сутки для CP2K (метод DFTB3). Результаты полуэмпирического моделирования методом DFTB3 в пакете Amber показаны на рисунке 1. Рассчитанный энергетический барьер составлял  $13 \pm 1$  ккал/моль.



**Рис. 1.** Начальное (А) и конечное (В) состояния реакции гидроксирования, полученные визуализацией кадров траектории движения, рассчитанной с помощью описанного метода молекулярного моделирования. Розовым цветом показан субстрат 2-гидроксибифенил, оранжевым – кофактор FAD-C4a-гидропероксид, зелёным – каталитический остаток His48. (С) Энергетическая диаграмма реакции гидроксирования, рассчитанная методом DFTB3.

Предварительное моделирование с помощью более точных методов функционала плотности (PBE/6-311++G\*\*) подтвердило данную оценку энергии и показало превосходство программы CP2K над коммерческим пакетом квантовой химии Gaussian: скорость счёта в CP2K 7.0 (метод GPW/OT/PBE/6-311++G\*\*) составляла 0.4 пс/сутки (16 узлов/«walker») против 0.02 пс/сутки для комбинации из Gaussian 16 (метод PBE/6-311++G\*\*, GPU не использовались) и программы *sander* из AmberTools 19/Amber 18 (Amber не имеет встроенных методов квантовой химии высокого теоретического уровня, но поддерживает расчёт во внешних программах). Поскольку интерфейс Gaussian в Amber использует Gaussian только на одном узле по модели SMP, на «walker» в режиме Amber/Gaussian выделялось по 2 узла (один для

Gaussian и один для Amber, лимитирующей стадией является Gaussian). В дальнейшей работе планируется продолжить расчёты методами высокого теоретического уровня с помощью описанной GPU/MPI-ускоренной версии CP2K и получить набор репрезентативных траекторий с более полным исследованием фазового пространства для уточнения энергетической диаграммы.

## Литература

1. Kanteev M. et al. A crystal structure of 2-hydroxybiphenyl 3-monooxygenase with bound substrate provides insights into the enzymatic mechanism // *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*. 2015. Vol. 1854. No. 12. P. 1906-1913. DOI: 10.1016/j.bbapap.2015.08.002
2. Bregman-Cohen A. et al. Altering 2-Hydroxybiphenyl 3-Monooxygenase Regioselectivity by Protein Engineering for the Production of a New Antioxidant // *ChemBioChem*. 2018. Vol. 19. №. 6. P. 583-590. DOI: 10.1002/cbic.201700648
3. Hutter J. et al. cp2k: atomistic simulations of condensed matter systems // *Wiley Interdisciplinary Reviews: Computational Molecular Science*. 2014. Vol. 4. №. 1. P. 15-25. DOI: 10.1002/wcms.1159
4. Salomon-Ferrer R., Case D. A., Walker R. C. An overview of the Amber biomolecular simulation package // *Wiley Interdisciplinary Reviews: Computational Molecular Science*. 2013. Vol. 3. №. 2. P. 198-210. DOI: 10.1002/wcms.1121
5. Tribello G. A. et al. PLUMED 2: New feathers for an old bird // *Computer Physics Communications*. 2014. Vol. 185. №. 2. P. 604-613. DOI: 10.1016/j.cpc.2013.09.018

## Использование вычислений в пониженной точности при реализации метода квантовых траекторий<sup>1</sup>

С.В. Боганов, В.Д. Волокитин, И.Б. Мееров

Нижегородский государственный университет им. Н. И. Лобачевского

Исследование динамики открытых квантовых систем является вычислительно трудоемкой задачей. В качестве основного уравнения, описывающего динамику таких систем, используют уравнение Линдблада [1]. Данное уравнение является матричным дифференциальным уравнением относительно матрицы плотности системы. Для численного исследования динамики больших открытых квантовых систем широко применяется метод квантовых траекторий [2, 3].

Метод квантовых траекторий заключается в усреднении по многим реализациям результатов интегрирования во времени волновой функции с помощью неэрмитовых гамильтонианов специального вида [4]. Данный метод использует идеологию метода Монте-Карло, что приводит к практически идеальному распараллеливанию для суперкомпьютеров как на общей, так и на распределенной памяти. Вместе с тем, известно, что получение релевантных результатов требует расчета большого числа траекторий. Кроме того, исследование моделей с большим числом состояний требует существенных ресурсов оперативной памяти для хранения промежуточных результатов вычислений. В этой связи представляет интерес вопрос об использовании вычислений с плавающей запятой одинарной и, в будущем, половинной точности для интегрирования во времени волновой функции, что занимает практически все время расчета. Потенциально возможным является и проведение расчетов в смешанном режиме, когда лишь критичные к точности вычисления выполняются в двойной точности, а другие – в одинарной или половинной точности, для ускорения работы программы и/или сокращения объема потребляемой памяти. Известно, что аналогичные подходы достаточно хорошо проработаны и используются для численного моделирования в ряде прикладных областей [6, 7, 8]. В данной работе мы делаем первый шаг в исследовании вопроса о возможности применения вычислений с плавающей запятой в арифметике пониженной точности для исследования динамики квантовых систем методом квантовых траекторий.

Для изучения рассмотренных возможностей в качестве тестовой задачи рассматривается широко распространенная модель – квантовый димер, тщательное исследование которого было выполнено в работе [2], а в качестве эталонной версии кода – программа, разработанная и проверенная при исследовании квантовых систем в работе [2] и ее продолжениях. Внесенные нами изменения в код позволили переключаться между двойной, одинарной и половинной точностью и провести первые вычислительные эксперименты для анализа погрешности вычислений и производительности/масштабируемости кода.

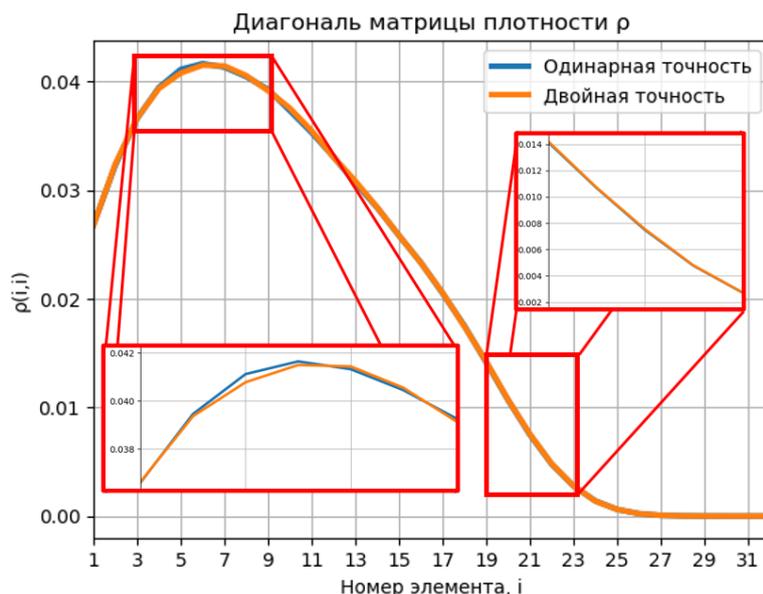
В проведенных экспериментах для задачи небольшого размера с  $N = 32$  состояниями было получено, что результаты в двойной и одинарной точности, полученные после усреднения одинакового – достаточно большого для сходимости метода Монте-Карло числа траекторий – хорошо согласуются (относительная ошибка менее 10%). На рисунке 1 продемонстрированы полученные значения диагональных элементов матрицы плотности на аттракторе системы. Как видно из графиков, понижение точности расчетов интегрирования волновой функции до одинарной не привело к критичной деградации точности полученного решения. Этот результат является обнадеживающим и позволяет надеяться, что этот эффект сохранится при решении задач большей размерности, что является темой дальнейших исследований. Соответствующие результаты будут представлены на конференции.

Проведение расчетов в половинной точности с использованием общего алгоритма для двойной точности оказалось затруднительным из-за получения больших ошибок при выполнении операции суммирования. Тем не менее, мы рассматриваем возможность использования специальных схем уменьшения ошибок. Так, для проведения операции суммирования, которое встречается при вычислении произведения матрицы на вектор и при подсчете нормы вектора,

<sup>1</sup> Работа выполнена при поддержке гранта РФФИ и Правительства Нижегородской обл. № 18-41-520004.

можно использовать известные схемы суммирования с компенсацией, например, алгоритм Кэ-хэна [5]. Такие схемы увеличат число операций, но все еще будут иметь смысл с точки зрения уменьшения затрат памяти, и, возможно, сокращения времени счета на устройствах, в которых вычисления с половинной точностью поддерживаются аппаратно, в частности, на NVidia и Intel GPU, а также на будущих архитектурах.

Тестовая задача запускалась на СК Intel Endeavor с узлами следующей конфигурации: 2x Intel Xeon Platinum 8260L, 2.40 GHz, 48 ядер, 192 ГБ оперативной памяти. Для интегрирования 1 миллиона траекторий на 200 периодов системы размера 32 потребовалось ~90 ядро-часов для двойной точности и порядка 74 ядро-часов для одинарной точности. Эффективность масштабируемости составляет более 90%. Несмотря на ожидаемое уменьшение времени расчета при переходе от двойной точности к одинарной, из-за вычислительных ошибок, несмотря на использование одной и той же последовательности псевдослучайных чисел, количество квантовых скачков на траекторию в расчете одинарной точности оказалось больше. Данный факт объясняет такую небольшую разницу во времени расчета и является предметом дальнейших исследований, как и попытка получения приемлемых результатов в половинной точности.



**Рис. 1.** Диагональ матрицы плотности квантового димера  $N = 32$  после 200 периодов, посчитанная методом квантовых траекторий при использовании  $10^6$  реализаций, для одинарной и двойной точности

## Литература

1. Lindblad G. On the generators of quantum dynamical semigroups // Communications in Mathematical Physics. – 1976. – Т. 48. – №. 2. – С. 119-130.
2. Volokitin V. et al. Computation of the asymptotic states of modulated open quantum systems with a numerically exact realization of the quantum trajectory method // Physical Review E. – 2017. – Т. 96. – №. 5. – С. 053313.
3. Kapit E. et al. Entanglement and complexity of interacting qubits subject to asymmetric noise // arXiv preprint arXiv:1905.01792. – 2019.
4. Mølmer K., Castin Y., Dalibard J. Monte Carlo wave-function method in quantum optics // JOSA B. – 1993. – Т. 10. – №. 3. – С. 524-538.
5. Higham N. J. The accuracy of floating point summation // SIAM Journal on Scientific Computing. – 1993. – Т. 14. – №. 4. – С. 783-799.
6. Clark M. A. et al. Solving Lattice QCD systems of equations using mixed precision solvers on GPUs // Computer Physics Communications. – 2010. – Т. 181. – №. 9. – С. 1517-1528.
7. Einkemmer L. A mixed precision semi-Lagrangian algorithm and its performance on accelerators // 2016 International Conference on HPC & Simulation (HPCS). – IEEE, 2016. – С. 74-80.

8. Chen G., Chacón L., Barnes D. C. An efficient mixed-precision, hybrid CPU–GPU implementation of a nonlinearly implicit one-dimensional particle-in-cell algorithm // *Journal of Computational Physics*. – 2012. – Т. 231. – №. 16. – С. 5374-5388.

## Новый курс по параллельной обработке графов\*

А.Ю. Пирова

Нижегородский государственный университет им. Н.И. Лобачевского

В ННГУ разработан новый курс, направленный на изучение параллельных алгоритмов для обработки больших графов. Курс входит в представленную ранее магистерскую программу «Вычислительные методы и суперкомпьютерные технологии» [1], целью которой является обучение специалистов, способных применять суперкомпьютерные технологии для современных междисциплинарных исследований и промышленных проектов. Программа охватывает различные области вычислительной математики и современных вычислительных технологий, в том числе углубленные курсы по структурам данных, параллельному программированию, машинному обучению и алгоритмам на графах.

Представление данных в виде графа используется во многих прикладных областях (биологические, транспортные, коммуникационные сети, энергетические системы, веб-графы и многие другие приложения). Выделяется ряд типичных задач по обработке графов, для которых требуются эффективные параллельные реализации (например, обход графа, поиск кратчайших путей, разделение графа и др.). Однако обработка больших объемов таких данных на современных вычислительных системах имеет определенные особенности [2]. Алгоритмы на графах относятся к «memory bound problems», то есть задачам, параллелизм которых ограничен из-за работы с памятью. Так, для них характерна низкая арифметическая интенсивность: число операций обращения к памяти значительно превышает число арифметических операций над данными. Большое число графов из приложений имеет нерегулярную структуру, что приводит к случайным доступам в массивы данных, значительному числу кэш-промахов, отсутствию векторизации. Кроме того, большое число алгоритмов формулируется в итерационном виде с зависимостью по данным между итерациями. Это затрудняет построение параллельного алгоритма и планирование вычислительной нагрузки, поскольку она не известна заранее.

Разработке эффективных реализаций графовых алгоритмов для высокопроизводительных систем уделяется большое внимание в научном и техническом сообществе, что подтверждается большим числом научных публикаций, разработкой новых библиотек. В последние годы в ведущих университетах мира студентам предлагаются курсы, посвященные этой теме. Например, курс проф. Дж. Шана в MIT [5], курс М. Джаффари в ETH Zürich [6], курс проф. Дж. Слота в RPI [7]. Также подобный спецкурс читается на ВМК МГУ А.С. Семеновым [8]. Разработка представленного в данной работе курса опирается на опыт выполнения научных проектов, связанных с вычислениями на графах, которые выполнялись в институте ИТММ ННГУ в течение последних лет. Так, авторами курса ведется разработка инструментов для решения разреженных систем линейных уравнений, в том числе, параллельных алгоритмов для переупорядочения графов разреженных матриц [3]. Алгоритмы, применяемые в этой задаче, также используются в задаче разделения графов. Коллектив института принимал участие в исследованиях молекулярных механизмов адаптации центральной нервной системы при воздействии ишемических повреждений головного мозга (например, синдром Дауна). В данной задаче применялись методы сетевого и статистического анализа для исследования метаболизма в сетях астроцитов [4].

Структура курса построена следующим образом. Во вводной части курса дается обзор типовых прикладных областей, в которых возникают графы большого порядка, а также обзор типовых задач. Обсуждаются основные особенности параллельных алгоритмов на графах, форматы хранения. Следующая лекция посвящена моделированию графов, в ней обсуждаются основные модели случайных графов и графов сетей, а также способы их программной реализации. В основной части курса рассматриваются классические задачи на графах: обход графа, поиск кратчайших путей из одной вершины, разделение графа, раскраска графа. Далее обсуждаются типовые алгоритмы для веб-графов и графов социальных сетей. В каждой лекции большое

---

\* А. Пирова поддержана стипендией Президента Российской Федерации, Приказ Минобр. науки РФ № 231 от 03.04.2018.

внимание уделяется обсуждению типичных особенностей реализации и оптимизации алгоритмов для вычислительных систем с общей и распределенной памятью. Приводятся примеры реализаций и результаты вычислительных экспериментов, показывающие эффективность обсуждаемых решений. В заключительной лекции дается сравнение библиотек, предназначенных для обработки больших графов. В практической части курса студентам предлагается выполнить реализацию одного из алгоритмов, провести анализ производительности и масштабируемости.

Основная литература по курсу включает источники [9–14]. Также к каждой теме дается список значимых научных публикаций за последние годы.

В ходе курса студенты изучат типовые подходы к построению параллельных алгоритмов для обработки графов, основы алгоритмической оптимизации, получат навыки работы с научной литературой и анализа результатов вычислительных экспериментов. По итогам прохождения курса студенты смогут как выполнять программную реализацию типовых параллельных алгоритмов на графах, так и разрабатывать новые параллельные алгоритмы для других задач.

## Литература

1. Meyerov I. et al. Bridging the Gap Between Applications and Supercomputing: A New Master's Program in Computational Science // Russian Supercomputing Days. Springer, Cham, 2019. P. 529–541. DOI: 10.1007/978-3-030-36592-9\_43
2. Lumsdaine A. et al. Challenges in parallel graph processing // Parallel Processing Letters. 2007. Vol. 17, No. 01. P. 5–20. DOI: 10.1142/S0129626407002843
3. Pirova A., Meyerov I., Kozinov E., Lebedev S. PMORSy: parallel sparse matrix ordering software for fill-in minimization // Optimization Methods and Software. 2017. Vol. 32, No. 2. P. 274–289. DOI: 10.1080/10556788.2016.1193177
4. Kustikova, V. et al. CalciumCV: computer vision software for calcium signaling in astrocytes // Lecture Notes in Computer Science. 2018. P. 168–179. DOI: 10.1007/978-3-030-11027-7\_17
5. Shan J. Course on Graph Analytics. <https://people.csail.mit.edu/jshun/6886-s18> (дата обращения: 10.04.2020)
6. Ghaffari M. Course on Algorithms for Large-Scale Graph Processing <http://www.vvz.ethz.ch/Vorlesungsverzeichnis/lerneinheit.view?lerneinheitId=129658&semkez=2018W&lang=en> (дата обращения: 10.04.2020)
7. Slota G.M. Course on Parallel Graph Analysis. <http://www.cs.rpi.edu/~slotag/classes/FA17/index.html> (дата обращения: 10.04.2020)
8. Семенов А.С. Курс «Параллельная обработка больших графов» [https://www.dislab.org/ru/PLGP\\_course](https://www.dislab.org/ru/PLGP_course) (дата обращения: 10.04.2020)
9. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 3-е издание. М.: «Вильямс», 2013. 1328 с.
10. Седжвик Р. Фундаментальные алгоритмы на C++. Алгоритмы на графах: Пер. с англ./Роберт Седжвик // СПб: ООО «ДиаСофтЮП», 2002.
11. Grama A. et al. Introduction to parallel computing. Pearson Education, 2003.
12. Roosta S. H. Parallel Processing and Parallel Algorithms: Theory and Computation. Springer, 2000. 566 p. DOI: 10.1007/978-1-4612-1220-1.
13. Newman M. Networks. Oxford university press, 2018.
14. Гергель В.П. и др. Высокопроизводительные параллельные вычисления. 100 заданий для расширенного лабораторного практикума. Учебное пособие. М.: ООО "Физматлит", 2018. 248 с.

## Опыт применения механизма областей для поэтапного распараллеливания программных комплексов с помощью системы SAPFOR \*

А.С. Колганов<sup>1</sup>

<sup>1</sup>Институт прикладной математики им. М.В. Келдыша РАН

Одна из основных сложностей разработки параллельной программы для кластера – необходимость принятия глобальных решений по распределению данных и вычислений с учетом свойств всей программы, а затем выполнения кропотливой работы по модификации программы и ее отладки. Большой объем программного кода, а также многомодульность, многовариантность и многоязыковость, затрудняют принятие решений по согласованному распределению данных и вычислений. Опыт использования предыдущей системы САП-ФОР показал, что при распараллеливании на кластер больших программ и программных комплексов необходимо уметь распараллеливать их постепенно, начиная с наиболее время-емких фрагментов и постепенно добавляя новые фрагменты, пока не достигнем желаемого уровня эффективности параллельной программы. С этой целью предыдущая система была полностью переработана, и на ее основе была создана новая система SAPFOR.

Система SAPFOR (System FOR Automated Parallelization) использует автоматически распараллеливающий компилятор для систем с распределенной памятью, который переводит последовательную пользовательскую программу в параллельную программу в модели DVMH [1]. Взаимодействие с пользователем осуществляется и на этапе подготовки последовательной программы к распараллеливанию, и в процессе этого распараллеливания.

Описываемая система автоматизированного распараллеливания ориентирована в большей степени на преобразование уже существующих последовательных программ в эффективные параллельные в модели DVMH. Но ее также успешно можно применить к программам, которые написаны «с нуля» в потенциально параллельном виде.

Данные программы могут содержать в себе отдельные модули, решающие определенные задачи, например, модуль ввода/вывода, модуль решения систем линейных уравнений, модуль, отвечающий за подготовительные работы. В большинстве случаев такие программы не нуждаются в полном распараллеливании, а в некоторых случаях полное распараллеливание попросту невозможно без существенного переписывания кода. Обычно бывает достаточно распараллелить только модули, производящие сложные вычисления над уже подготовленными данными. С другой стороны, программа может быть настолько большой, что программисту необходимо понять, можно ли вообще получить эффективно работающую программу на конкретной архитектуре. Для выполнения такой оценки достаточно попытаться распараллелить только некоторые модули такой программы, которые могут занимать до 90% времени.

Для обеспечения возможности инкрементального распараллеливания на кластер в системе SAPFOR было введено понятие области распараллеливания [2, 3]. Это позволяет последовательно переходить от рассмотрения отдельных небольших областей к более крупным областям, вплоть до целой программы, сохраняя при этом преемственность ранее принятых решений по распараллеливанию отдельных областей и уточняя их при необходимости. Области распараллеливания могут быть построены автоматически на основе времен, полученных с помощью профилирования последовательной программы, либо заданы вручную пользователем. Области отражают те участки кода исходной программы, которые будут рассматриваться системой SAPFOR.

---

\*Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов 18-01-00851, 19-07-00889, 19-07-01101, 20-01-00631.

Использование механизма областей распараллеливания хорошо подходит в тех случаях, когда программист сталкивается с незнакомой ему программой. Чем больше рассматриваемая программа, тем сложнее процесс ее распараллеливания на кластер, так как необходимо представлять структуру программы, знать ее поведение при различных входных данных. Конечная цель – отображение всей программы на кластер с графическими процессорами.

Механизм областей был успешно применен программе Block Tridiagonal Solver, решающей 3-х мерную систему уравнений Навье-Стокса для сжимаемой жидкости или газа. В данной программе используется трехдиагональная схема и метод переменных направлений. Исходная версия данной программы содержит 3200 строк кода в фиксированном формате Фортран, 17 файлов и 19 процедур.

В результате профилирования, было выяснено, что 95% времени занимают 40% от всего кода. С помощью системы SAPFOR было выполнено распараллеливание программы «снизу в вверх» – начиная от самых вложенных процедур и до всей программы целиком. Система SAPFOR позволила не только провести анализ программы, но и выполнить одно из необходимых для распараллеливания преобразований – «расширение приватизируемых переменных». Преобразования выполняются автоматически по указанию пользователя с помощью расстановки директив системе через диалоговую оболочку – визуализатор системы SAPFOR. В результате чего, полученная версия программы смогла ускориться на одном GPU Tesla V100 в 16.5 раз по сравнению с исходной последовательной версии (на наборе данных C).

Для эффективного выполнения вычислительных регионов на нескольких GPU требуется дополнительная оптимизация, которая позволит получить еще больший выигрыш при использовании гетерогенного кластера. Проведенные вручную оптимизации полученной после системы SAPFOR программы, позволили получить ускорение на одном GPU Tesla V100 в 123 раза, на двух GPU – в 170 раз и на четырех GPU – в 221 раз (на наборе данных D). Распараллеленная авторами программа в модели MPI смогла ускориться в 159 раз на 256 процессах кластера K60 (установленного в ИПМ им. М. В. Келдыша РАН).

Благодаря введенному в системе SAPFOR инкрементальному подходу по распараллеливанию программ удастся сначала сузить область действия системы на наиболее время затратные участки кода, а затем распространить полученные решения на всю программу, если это возможно. На примере распараллеливания описанной выше программы, можно сделать следующие выводы. В отличие от распараллеливания с использованием стандартных технологий и расширений MPI, OpenMP и CUDA, разработка, преобразования и поддержка кода осуществляются программистом в последовательной программе. Полученная при этом DVMH-программа с помощью системы SAPFOR может задействовать многоядерные процессоры и графические ускорители, что позволит тратить меньше вычислительных ресурсов кластера с большей эффективностью.

## Литература

1. DVM–система [Электронный ресурс]. URL: <http://dvm-system.org/> (дата обр. 01.01.2020).
2. Колганов А. С., Яшин С. В. Автоматическое инкрементальное распараллеливание больших программных комплексов с помощью системы SAPFOR. Труды Международной Параллельные вычислительные технологии (ПаВТ'2019), Издательский центр ЮУрГУ Челябинск, 2019, С. 275-287.
3. Бахтин В. А., Колганов А. С., Крюков В. А. и др. Инкрементальное распараллеливание для кластеров в системе САПФОР. Труды Всероссийской научной конференции «Научный сервис в сети Интернет». 2017, С. 48-52. DOI: 10.20948/abrau-2017.

## Особенности трубок чувствительности волн РКІКР и РКІКР во внутреннем ядре Земли

О.А. Усольцева, В.М. Овчинников

Институт динамики геосфер им. М.А. Садовского РАН

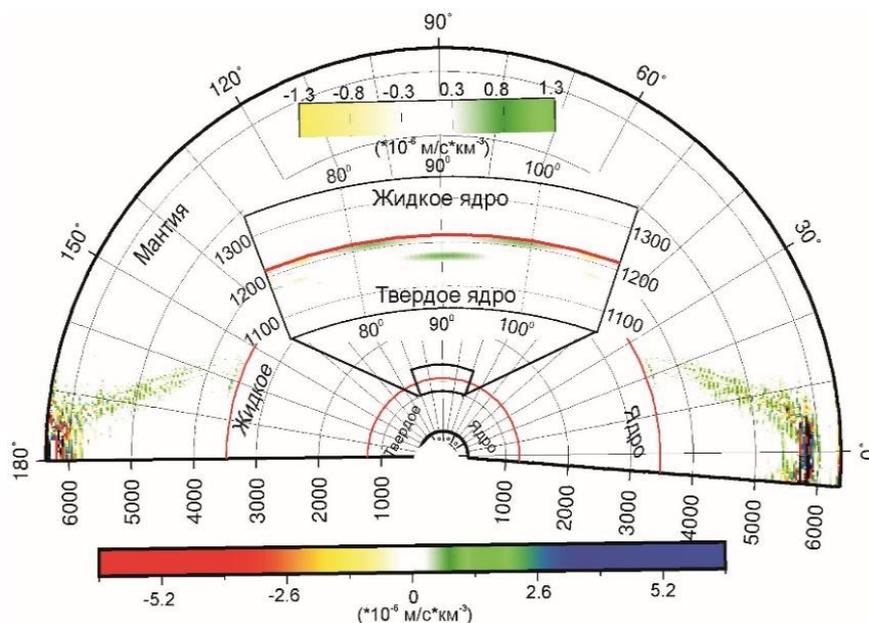
В последние 20 лет внутреннее ядро Земли является объектом активных сейсмологических исследований, прежде всего из-за его значительной роли в происхождении и эволюции магнитного поля Земли. Скорость поперечных волн - одно из наиболее слабо изученных свойств внутреннего ядра. В недавних работах [1, 2] для решения этой задачи начали использовать волны РКІКР (однократно отраженные от границы (ICB) между внешним и внутренним ядром волны со стороны внутреннего ядра), амплитуды которых чувствительны к скорости поперечных волн. Из этих исследований следует, что верхняя часть внутреннего ядра толщиной около 60 км, которая зондируется волнами РКІКР, имеет латерально неоднородное распределение скорости поперечных волн, которое может быть проинтерпретировано как неоднородность теплового потока на границе ICB, влияющего на характер конвекции во внешнем ядре. Однако надежность этих выводов требует дополнительных независимых исследований, так как амплитуды волн РКІКР очень малы.

В настоящем исследовании мы приводим результаты, опирающиеся на расчет ядер чувствительности, т. е. функционалов, которые определяют чувствительность отдельных измерений к изменению механических параметров среды в различных пространственных областях, во внутреннем ядре для волн РКІКР (прошедших через внутреннее ядро) и РКІКР. Для вычисления функционалов использован метод, предложенный в [3] и его программная реализация DSM Kernel Suite [4] (<https://github.com/IPGP/DSM-Kernel>) на языке Фортран с использованием MPI, позволяющая вычислить ядра чувствительности на эпицентральных расстояниях до  $180^\circ$  для волн с минимальным периодом 1с. В вычислительном плане задача состоит из двух этапов. На первом для стандартной модели Земли ak135 с источником сейсмических возмущений на глубине 557 км рассчитывается база тензоров напряжения Грина для сейсмограмм длительностью 1638с. На втором этапе для конкретного источника S и приемника R и диапазона частот F (например, S:  $0^\circ$  с.ш. и  $0^\circ$  в.д., R:  $0^\circ$  с.ш. и  $180^\circ$  в.д., F: 0.1-0.5 Гц) проводится расчет различных ядер чувствительности: ядер вариаций времен пробега волн РКІКР и РКІКР от скорости поперечных волн  $V_s$  и скорости продольных волн  $V_p$ , ядер вариаций их амплитуд от  $V_s$ ,  $V_p$ , а также параметров затухания Q.

Время расчета первой части на 128 процессорных ядрах с частотой процессоров 2.9 ГГц для сетки с шагом по азимуту  $0.5^\circ$  и радиусу 20 км и минимальным возможным периодом 2с равно 4 часа 20 минут. Для распараллеливания первый процессор с помощью функции широковещательной рассылки MPI\_Bcast отправляет остальным информацию о входных параметрах. В ходе цикла каждый процессор последовательно проводит однотипные вычисления для шести (семи) различных частот. В итоге получается база из тензоров напряжения Грина объемом 80 Гб (для сравнения в [3] на 20-ти ядерном компьютере с частотой процессоров 2.5 ГГц время расчета ядер чувствительности для сейсмических волн в мантии составило более 1000 часов). Вторая часть расчетов по времени занимает около 19 минут на 128 процессорных ядрах.

Во внутреннем ядре для волны РКІКР и эпицентрального расстояния  $180^\circ$  максимальная чувствительность составляет  $1.5 \cdot 10^{-6}$  с/км<sup>3</sup>, а ширина первой зоны Френеля 500 км. Для волны РКІКР в верхней части внутреннего ядра максимальная чувствительность  $1 \cdot 10^{-6}$  с/км<sup>3</sup>, а в основании внешнего  $15 \cdot 10^{-6}$  с/км<sup>3</sup>. Таким образом, на время пробега волны РКІКР при равном объеме неоднородности в основании внешнего ядра оказывают значительно большее влияние, чем неоднородности во внутреннем ядре. Но, по-видимому, основной результат следует из анализа амплитудных ядер чувствительности (рис.1) для временного интервала сейсмограммы с волной РКІКР: неоднородность скорости поперечных волн в 1% в области источника сейсмических возмущений при условии равного объема создает в 10 раз большие кросс-

корреляционные вариации амплитуды  $\delta A_{\text{cross-cor}} = (A1^{\text{max}} - A2^{\text{max}}) / A2^{\text{max}}$ , ( $A1^{\text{max}}$  - максимальная амплитуда кросс-коррелограммы между рассчитанной сейсмограммой и реальной,  $A2^{\text{max}}$  - максимальная амплитуда автокоррелограммы рассчитанной сейсмограммы), чем 1% неоднородность скорости поперечных волн, локализованная в верхней части внутреннего ядра. Этот результат указывает на необходимость более критического и тщательного отношения к выводам о структурных особенностях и скорости поперечных волн, полученных из отношения амплитуд волн РКПКР/РКИКР.



**Рис. 1** Амплитудное ядро чувствительности в зависимости от скорости поперечных волн  $V_s$  для фрагмента сейсмограммы, включающего волну РКПКР на эпицентральной дистанции  $180^\circ$ , и для волн с минимальным периодом 2с.

Исследование выполнено при финансовой поддержке Российского Фонда фундаментальных исследований – грант № 18-05-00619 и с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М.В.Ломоносова [5].

## Литература

1. Attanayake J, Thomas C., Cormier V. F., Miller M. S., Koper K. D.. Irregular Transition Layer Beneath the Earth's Inner Core Boundary From Observations of Antipodal PKIKP and PKIKP Waves // 3G. 2018. doi: 10.1029/2018GC007562.
2. Краснощеков Д.Н., Овчинников В.М., Усольцева О.А. О скорости поперечных волн в вершине внутреннего ядра Земли // ДАН. 2019.т.488. №4, с.434-438.
3. Zhao L., Chevrot S. An efficient and flexible approach to the calculation of three-dimensional full-wave Fréchet kernels for seismic tomography—I. Theory // GJI. 2011, 185, 922–938
4. Fuji N., Chevrot S., Zhao L., Geller R. J., Kawai K. Finite-frequency structural sensitivities of short-period compressional body waves // GJI, 2012, 190, 522–540 doi: 10.1111/j.1365-246X.2012.05495.x
5. Воеводин В. , Жуматий С., Соболев С., Антонов А., Брызгалов П., Никитенко Д., Стефанов К., Воеводин В. Практика суперкомпьютера "Ломоносов" // Открытые системы, N 7, 2012. С. 36-39.

## Применение технологий машинного обучения для оптимизации параметров методов решения систем уравнений\*

А.А. Петрушов, Б.И. Краснопольский

НИИ механики МГУ

Решение систем линейных алгебраических уравнений (СЛАУ) является часто встречающейся подзадачей в задачах математической физики. Разнообразие дифференциальных уравнений и сопутствующих методов их численного решения породило обширный набор методов решения систем уравнений, учитывающих те или иные особенности матрицы СЛАУ.

Наиболее простыми с точки зрения практического применения являются прямые методы решения СЛАУ. Данные методы содержат минимальное количество настроечных параметров. Однако, их применение на практике обычно ограничивается простыми геометриями расчетных областей (когда оказывается возможным использование прямых быстрых методов на основе быстрого преобразования Фурье или методов циклической редукции), СЛАУ с матрицами специального вида или системами уравнений с плохообусловленными матрицами, для которых не удастся построить эффективный предобуславливатель. При использовании сеточных методов дискретизации исходных дифференциальных уравнений, указанные выше критерии зачастую не выполняются. Для решения соответствующих систем уравнений более эффективным оказывается применение итерационных методов общего вида, например, итерационных методов подпространства Крылова с предобуславливателями на основе методов неполного LU-разложения [1] или многосеточных методов [2].

Если методы подпространства Крылова обычно не имеют настроечных параметров, то для указанных методов предобуславливания имеется определенное количество управляющих параметров, существенно влияющих на скорость сходимости и время решения системы уравнений. Диапазоны оптимальных значений параметров существенно зависят от свойств матрицы СЛАУ, и их подбор необходимо выполнять для каждой конкретной задачи. В случае методов неполного LU-разложения число таких параметров измеряется единицами, что оставляет возможность проведения такого поиска вручную. Однако, количество управляющих параметров для многосеточных методов измеряется несколькими десятками, что делает на практике невозможным проведение полноценного поиска, и требует априорных знаний о близких к оптимальным значениях для ряда настроечных параметров.

Достаточно распространенной в прикладных расчетах оказывается ситуация, когда в ходе длительных нестационарных расчетов необходимо решать СЛАУ с неизменной во времени матрицей, но постоянно меняющейся правой частью. Примером такой ситуации являются вихреразрешающие методы моделирования несжимаемых турбулентных течений, когда на каждом шаге по времени возникает потребность решения уравнения Пуассона для давления (одного или нескольких), а количество таких шагов интегрирования может быть порядка  $10^5 - 10^6$ . Таким образом, изначальные затраты времени на подбор параметров численного метода, сопоставимые с несколькими десятками или сотнями решений СЛАУ, в масштабах всего расчета могут быть оправданными, с одной стороны, обеспечив ускорение, а с другой – частично автоматизировав и упростив процедуру расчетов.

В настоящей работе исследуется возможность применения технологий машинного обучения для оценки времени решения СЛАУ и поиска оптимальных параметров алгоритма решения. На данном этапе акцент делается на решении СЛАУ, полученных при дискретизации эллиптических уравнений, как наиболее востребованных в проводимых нами расчетах турбулентных течений [3]. В дальнейшем планируется рассмотреть возможность обобще-

\*Работа поддержана грантом РФФИ 18-71-10075.

ния результатов на другие типы уравнений. В [4, 5] и ряде других работ рассмотрен подход на основе машинного обучения, позволяющий по виду исходной матрицы системы оценить время решения системы уравнений для predetermined набора методов решения СЛАУ. В отличие от указанных выше, настоящая работа ставит более локальную задачу, а именно – подбор параметров конкретного набора методов для определенного типа матриц.

В качестве метода решения СЛАУ в настоящей работе использовался метод BiCGStab с предобуславливателем на основе классического алгебраического многосеточного метода. В общем случае, данная комбинация методов имеет несколько десятков параметров. Для проведения первых экспериментов и отладки инструментария в целом, было выбрано 9 целевых параметров, тогда как остальные были зафиксированы. В качестве базовых использованы две СЛАУ для эллиптических уравнений: модельная система для равномерной сетки  $35^3$  в кубической расчетной области и СЛАУ из расчета задачи о моделировании турбулентного течения в плоском канале с кубическим препятствием на стенке [3] (2.32 млн. неизвестных).

Для предсказания времени решения СЛАУ модельной задачи была построена полносвязная нейронная сеть с тремя слоями (256, 128 и 32 нейрона). На вход подавались значения целевых параметров для оптимизации, тогда как на выходе формировалось предполагаемое время решения. В процессе обучения настраивались основные параметры модели, такие как функции активации, шаг оптимизатора, нормировка данных, и другие. По итогам обучения данной сети на 1000 расчетов удалось достичь точность предсказания времени решения СЛАУ порядка 93%. На основе модели построены и исследованы зависимости времени решения от различных пар параметров алгоритма решения. Найдены минимумы для времени решения и соответствующие им значения в пространстве входных параметров.

Аналогичная нейронная сеть построена и обучена для матрицы 2.32 млн. неизвестных. Результаты предсказания сети, построенной для большей матрицы, планируется сопоставить с результатами экстраполяции сети, построенной для малой тестовой матрицы, с помощью технологии переноса обучения (transfer learning). Также планируется разработка подхода для обобщения результатов для небольших матриц на более практически значимые размеры задач одного семейства исходных дифференциальных уравнений.

Целью данной работы является создание компактного инструментария для оценки влияния различных параметров многосеточных методов на скорость сходимости и время решения СЛАУ, и выделение редуцированного набора параметров, подлежащих оптимизации при проведении расчетов. В дальнейшем, в случае успеха начатой работы, разрабатываемый инструментарий для предсказания оптимального набора параметров будет внедрен в разрабатываемую библиотеку **XAMG** для решения СЛАУ со многими правыми частями.

## Литература

1. Saad Y. Iterative Methods for Sparse Linear Systems, Second Edition. SIAM, 2003. 528 p.
2. Trottenberg U., Oosterlee C.W., Schuller A. Multigrid. Academic Press, 2001. 631 p.
3. Krasnopolsky B. An Approach for Accelerating Incompressible Turbulent Flow Simulations Based on Simultaneous Modelling of Multiple Ensembles // Computer Physics Communications. 2018. Vol. 229. P. 8-19.
4. Bhowmick S., Eijkhout V., Freund Y., Fuentes E., Keyes D. Application of Machine Learning in Selecting Sparse Linear Solvers // Tech.report. 2006.  
<https://pages.tacc.utexas.edu/~eijkhout/Articles/2006-bhowmick.pdf>
5. Yeom J.-S., J. Thiagarajan J., Bhatele A., Bronevetsky G., Kolev T. Data-Driven Performance Modeling of Linear Solvers for Sparse Matrices // Proceedings of 7th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS). 2016. P. 32–42.

## Реализация метода конечных разностей во временной области с использованием вычислений в двойной, одинарной и половинной точности<sup>1</sup>

А. Н. Арисова, В. Д. Волокитин, И. Б. Мееров

Нижегородский государственный университет им. Н. И. Лобачевского

Стремительное развитие вычислительных систем открывает новые возможности для численного моделирования. Одно из актуальных направлений последних лет связано с появлением в различных устройствах аппаратных реализаций чисел с плавающей запятой половинной точности. В этой связи представляет интерес изучение возможности использования такой арифметики в классических задачах численного моделирования. На первый взгляд может показаться, что об этом не может быть и речи. Действительно, во многих задачах требуется аккуратная реализация даже при использовании двойной точности. Тем не менее, опыт последнего десятилетия показывает, что в некоторых приложениях возможна разработка и применение определенных подходов для организации вычислений с использованием одинарной, половинной точности и их комбинации, что в будущем позволит экономить память и сокращать время расчета.

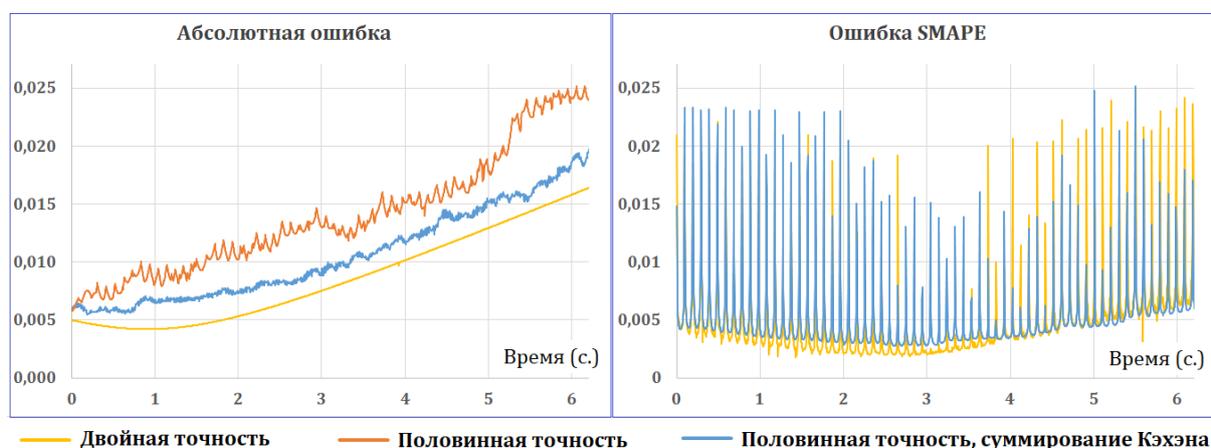
Перспективы данного направления исследований продемонстрированы и подтверждены в вычислительных экспериментах при решении классических задач линейной алгебры [1, 2], конечно-элементного анализа [3], при моделировании климата [4] и в других приложениях. Появившаяся аппаратная поддержка арифметики половинной точности в современных GPU наряду с перспективами появления такой поддержки в CPU придали новый импульс соответствующим исследованиям в линейной алгебре [5], при тренировке искусственных нейронных сетей [6], а также в вычислениях общего назначения [7]. В данном докладе этот вопрос изучается в контексте решения уравнений Максвелла методом конечных разностей во временной области [8, 9] (Finite-Difference Time-Domain, далее – FDTD).

Мы основывались на реализации, многократно апробированной ранее при моделировании лазерной плазмы методом частиц в ячейках программным комплексом PICADOR [10]. В качестве первого шага была реализована поддержка трехмерной прямой сетки с разностной аппроксимацией производной по пространству второго порядка и периодическими граничными условиями (упрощенный вариант метода). С использованием многоядерного CPU выполнялось моделирование распространения волны в вакууме в задаче с известным аналитическим решением. Проведены расчеты в двойной, одинарной и половинной точности, изучена зависимость абсолютной и относительной ошибки, а также интегрированной по пространству ошибки SMAPE в зависимости от шага интегрирования и точности вычислений с плавающей запятой в сравнении с аналитическим решением. Первые результаты показали, что в достаточно простых модельных задачах использование вычислений с плавающей запятой в одинарной точности является вполне приемлемым, тогда как при переходе к вычислениям в половинной точности проявляется проблема потери значащих цифр мантииссы при выполнении операции суммирования. Для преодоления этого эффекта был реализован алгоритм компенсации ошибок суммирования Кэхэна [11], показавший хорошие результаты (рис. 1). На графиках приведены вычисленные в сравнении с аналитическим решением значения абсолютной ошибки и интегрированной ошибки по пространству SMAPE при интегрировании на один период по времени на трехмерной сетке из 64 узлов по каждой координате. В связи с тем, что результаты, полученные в двойной точности и в одинарной точности, близки, последние опущены, чтобы не загромождать график. При вычислении ошибки SMAPE не приведены результаты вычислений в половинной точности без использования схем суммирования с компенсацией ошибок, т.к. выяснилось, что это не имеет практического смысла. Результаты показали, что при вычислениях в половинной точности удастся получить значения ошибок, близкие к ошибкам вычисления в двойной точности (разница ошибок ~10%), и, возможно, эти результаты могут быть улучшены.

<sup>1</sup> Работа выполнена при поддержке гранта РФФИ и Правительства Нижегородской обл. №18-47-520001.

Замеры производительности производились на СК Intel Endeavor 2x Intel Xeon Platinum 8260L, 2.40 GHz, 48 ядер, 192 ГБ оперативной памяти. Эффективность распараллеливания в рамках одного сокета во всех реализациях составляет выше 95%, при задействовании второго сокета эффективность падает до 90%. При этом времена работы программы в двойной и одинарной точности являются сопоставимыми, так как не проводилось оптимизаций, направленных на векторизацию кода. Половинная точность проигрывает в производительности в 5-6 раз, так как на данный момент отсутствует аппаратная поддержка данного типа в CPU и для ее эмуляции используется сторонняя библиотека Half [13].

В докладе будет представлен подробный анализ ошибок, а также продемонстрирована производительность и эффективность масштабируемости разработанных реализаций. Дальнейший анализ ошибок и поиск методов их уменьшения наряду с проверкой работоспособности анализируемых методов в практических приложениях является одним из направлений дальнейших исследований. В частности, предполагается реализация сетки Yee [8] и граничных условий PML [12] (возможно, расчеты граничных условий будут выполняться в двойной точности), с последующим решением классических задач электродинамики.



**Рис. 1.** Ошибки численного интегрирования уравнений Максвелла методом FDTD. Задача о распространении волны в вакууме. 1 период по времени. Сетка 64 x 64 x 64. Шаг по времени  $dt = 0.005$ .

## Литература

1. Baboulin M. et al. Accelerating scientific computations with mixed precision algorithms // Computer Physics Communications. – 2009. – Т. 180. – №. 12. – С. 2526-2533.
2. Buttari A. et al. Using mixed precision for sparse matrix computations to enhance the performance while achieving 64-bit accuracy // TOMS. – 2008. – Т. 34. – №. 4. – С. 1-22.
3. Göddeke D., Strzodka R., Turek S. Performance and accuracy of hardware-oriented native-, emulated-and mixed-precision solvers in FEM simulations // International Journal of Parallel, Emergent and Distributed Systems. – 2007. – Т. 22. – №. 4. – С. 221-256.
4. Dawson A. et al. Reliable low precision simulations in land surface models // Climate Dynamics. – 2018. – Т. 51. – №. 7-8. – С. 2657-2666.
5. Haidar A. et al. Harnessing GPU tensor cores for fast FP16 arithmetic to speed up mixed-precision iterative refinement solvers // SC18. – IEEE, 2018. – С. 603-613.
6. Courbariaux M., Bengio Y., David J. P. Training deep neural networks with low precision multiplications // arXiv preprint arXiv:1412.7024. – 2014.
7. Henry G., Tang P. T. P., Heinecke A. Leveraging the Bfloat16 artificial intelligence datatype for higher-precision computations // 2019 ARITH. – IEEE, 2019. – С. 69-76.
8. Yee K. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media // IEEE Tr. on ant. & prop. – 1966. – Т. 14. – №. 3. – С. 302-307.
9. Taflove A. Application of the finite-difference time-domain method to sinusoidal steady-state electromagnetic-penetration problems // IEEE Trans. on el. comp. – 1980. – №. 3. – С. 191-202.
10. Bastrakov S. et al. Particle-in-cell plasma simulation on heterogeneous cluster systems // Journal of Computational Science. – 2012. – Т. 3. – №. 6. – С. 474-479.

11. Higham N. J. Accuracy and stability of numerical algorithms. – Siam, 2002. – Т. 80.
12. Berenger J. P. et al. A perfectly matched layer for the absorption of electromagnetic waves // Journal of computational physics. – 1994. – Т. 114. – №. 2. – С. 185-200.
13. Half-precision floating-point library. URL: <http://half.sourceforge.net/>

## Симулятор системы управления суперкомпьютерными заданиями как научный сервис

А. Баранов, Д. Ляховец

Межведомственный суперкомпьютерный центр Российской академии наук

Системы управления заданиями являются неотъемлемым программным обеспечением для высокопроизводительных вычислительных систем коллективного использования [1]. СУЗ отвечает за такие функции, как хранение очереди заданий разных пользователей, определение порядка их запуска, выделение вычислительных ресурсов для запускаемых заданий, запуск заданий, контроль их завершения, проверку освобождения вычислительных ресурсов. Качество СУЗ оценивается по многим параметрам, таким как утилизация вычислительных ресурсов, среднее время ожидания задания в очереди и другие.

Современные СУЗ (такие, как SLURM, PBS, Moab, СУППЗ) развивались в течение десятилетий, эволюционируя в сложные комплексные системы со множеством настроек. Примером настроек СУЗ может выступать алгоритм планирования заданий и его параметры, способ учёта приоритета пользователей и групп, ограничение на размер запускаемых заданий в различное время. Определение оптимальных настроек СУЗ является сложной задачей, поскольку влияние этих настроек на показатели качества СУЗ не всегда очевидно, часто улучшение одного показателя влечет за собой ухудшение других.

Помимо определения оптимальных настроек СУЗ, актуальной задачей является прогнозирование времени запуска конкретного задания из очереди СУЗ. Это позволит пользователю получать более точную оценку времени запуска. В случае объединения нескольких СУЗ в территориально распределённую систему [2] прогнозирование запуска позволяет определять ту СУЗ, в которой задание будет обработано быстрее. На основе подобного прогноза может быть создана глобальная очередь заданий, объединяющая несколько СУЗ, что позволит осуществлять балансировку вычислительной работы путем перераспределения заданий из более загруженных систем в менее загруженные.

Интерес представляют различные аспекты моделирования СУЗ для решения двух поставленных задач: (1) поиска наилучших настроек СУЗ и (2) предсказания времени запуска заданий.

Проведённый авторами анализ работ по моделированию СУЗ (в том числе таких симуляторов СУЗ, как Alea, ScSF, Batsim, AccaSim, SLURM simulator) выявил две научные проблемы для дальнейшего исследования.

Первая проблема состоит в валидации модели, точнее – в отсутствии общепринятых способов оценки адекватности модели СУЗ и точности моделирования. В актуальных научных публикациях авторы либо игнорируют необходимость валидации модели, либо применяют методику визуального сравнения графиков, либо рассчитывают интервальные статистические показатели качества. Отсутствуют аналитические критерии определения меры точности или адекватности модели. Нами ведутся исследования в этом направлении, первые результаты представлены в работе [3].

Вторая проблема – сложность воспроизведения результатов моделирования других исследователей. Исследователи применяют различные модели, параметры и входные данные, а в качестве результата публикуют только часть выходных данных (например, интегральные показатели качества). Нередко применяемые модели, часть параметров, и полные входные и выходные данные недоступны публично. В результате нельзя полностью воспроизвести результаты моделирования других исследователей, что противоречит научному принципу воспроизводимости результатов.

Доклад посвящен проблеме воспроизведения экспериментов с моделями СУЗ. Авторы рассматривают новые способы применения симулятора СУЗ как общедоступного сервиса для исследователей. С помощью такого сервиса исследователь получает возможность проверять различные гипотезы о влиянии на показатели эффективности СУЗ, при этом (1) его исследование производится на стороне сервиса в круглосуточном режиме, (2) точность или адекватность ра-

боты симулятора обеспечивается сервисом, (3) обеспечивается воспроизводимость результатов исследования, и симулятор-как-сервис становится единой точкой входа для исследователей. Такой способ не ограничивает исследователей в применении собственных моделей для частного использования или создании множества альтернативных публичных моделей, но позволяет привлечь новых исследователей, снижая порог входа за счёт доступной модели для проверки различных настроек СУЗ.

Работа выполнена в МСЦ РАН в рамках проекта по гранту РФФИ № 18-29-03236.

## Литература

1. Reuther, A. et al.: Scalable system scheduling for HPC and big data. In: Journal of Parallel and Distributed Computing, 2018, vol. 111, pp. 76–92. DOI: 10.1016/j.jpdc.2017.06.009.
2. Баранов А.В., Тихомиров А.И., «Методы и средства организации глобальной очереди заданий в территориально распределенной вычислительной системе», Вестн. ЮУрГУ. Сер. Выч. матем. информ., 6:4 (2017), 28–42. DOI: 10.14529/cmse170403.
3. Baranov, A., Telegin, P., Shabanov, B. Lyakhovets, D.: Measure of Adequacy for the Supercomputer Job Management System Model. In: Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS, Vol. 18, pages 423–426 (2019). DOI: 10.15439/2019F186.

## Сравнение алгоритмов огрубления графа в задаче параллельного переупорядочения разреженной матрицы \*

А.Ю. Пирова, А.А. Оболенский, В.В. Девликамов, А.Ю. Нестеров

Нижегородский государственный университет им. Н.И. Лобачевского

Симметричное переупорядочение строк и столбцов – один из этапов прямого решения системы линейных уравнений с разреженной матрицей. Эта процедура позволяет снизить число ненулевых элементов в факторизованной матрице, что определяет время и потенциал параллелизма дальнейшей численной факторизации. В настоящее время доступен ряд специализированных пакетов для переупорядочения разреженных матриц. Так, широко используются открытые библиотеки Metis, ParMetis (<http://glaros.dtc.umn.edu/gkhome/views/metis>) и Scotch, PT-Scotch (<https://www.labri.fr/perso/pelegrin/scotch/>), предназначенные для систем с распределенной памятью. В 2015 г. был представлен переупорядочиватель для систем с общей памятью *ptmetis* [5]. Авторами данной работы была разработана библиотека PMORSy, в которой переупорядочение выполняется многоуровневым методом вложенных сечений. Параллелизм в PMORSy ориентирован на системы с общей памятью и основан на использовании механизма логических задач [1].

В многоуровневом методе вложенных сечений разделитель вычисляется в три этапа: огрубление, разделение, развертывание [2]. Выбор конкретных алгоритмов, используемых на каждом этапе, и их параметров для конкретной матрицы – отдельная задача исследователя. Как правило, выбирается некоторый компромиссный вариант между алгоритмами, дающими лучшее качество работы в смысле числа ненулевых элементов фактора матрицы, и алгоритмов, дающих приемлемое заполнение за наименьшее время работы. Целью данной работы было исследование новых для PMORSy алгоритмов огрубления с целью получения лучшего качества перестановок. Для этого была выполнена программная реализация алгоритма глобальных путей (Global Paths Algorithm, GPA) [3] и алгоритма линейной аппроксимации (Linear Time Approximation Algorithm, LAM) [4]. Реализация алгоритмов огрубления графов, использованных в работе, выполнена в виде отдельного модуля на языке C++. Общая процедура переупорядочения реализована на языке C с использованием механизма логических задач OpenMP 3.0 (*openmp task*).

Вычислительные эксперименты проводились на узлах кластера «Ломоносов» со следующими характеристиками: процессор 2×8 ядер Intel Xeon 5570 (8 ядер на узел), память 12 Gb, компилятор Intel Parallel Studio v. 2017.1. В качестве тестовых матриц были выбраны симметричные матрицы из коллекции Suite Sparse (<https://sparse.tamu.edu/>) размерностью от  $2 \times 10^5$  до 1 млн. строк. Проведено сравнение с библиотекой ParMETIS v. 4.0.3. Алгоритмы GPA и LAM сравнивались с ранее реализованными алгоритмами случайных паросочетаний (RM) и паросочетания тяжелых ребер (HM) [2]. Для алгоритма GPA в качестве вспомогательных алгоритмов использовались алгоритмы RM, HM и алгоритм паросочетания максимального веса (MWM). В процедуре переупорядочения граф сжимался по двум критериям останова: (а) не более 10 шагов, до 20 вершин; (б) не более 20 шагов, до 100 вершин. Таким образом, на вход алгоритмов, выполняющих этапы многоуровневого метода, в конфигурации (а) поступают графы меньшего размера, но более плотные, чем в конфигурации (б).

В таблице 1 показано сравнение заполнения факторов матриц, полученных в разных конфигурациях PMORSy, и ParMETIS. PMORSy получено меньшее заполнение, чем ParMETIS, для 3 матриц из 7, еще для двух матриц отставание в пределах 6%. Для графов, полученных при конечно-элементных расчетах, были получены близкие по качеству перестановки для всех алгоритмов в обеих конфигурациях (разница в пределах 4%). Лучшие результаты по качеству были получены алгоритмами HM и GPA(HM) для большинства матриц. При этом в конфигурации (б) факторы матриц в среднем на 5% больше, чем в конфигурации (а). Алгоритмы RM и HM

---

\* А. Пирова поддержана стипендией Президента Российской Федерации, Приказ Минобр. науки РФ № 231 от 03.04.2018.

показали близкое время работы, алгоритмы GPA(RM), GPA(HM) работают до 5 раз дольше. В конфигурации (б) время работы в среднем на 13% меньше, чем в конфигурации (а). В таблице 2 показано время работы PMORSy в сравнении с ParMETIS. Для каждой матрицы приведены данные для типа огрубления, дающего наименьшее время работы. На данном тестовом наборе PMORSy опережает ParMETIS на 5 матрицах. Для PMORSy было получено ускорение около 3 раз для всех рассматриваемых алгоритмов на матрицах размерностью более  $5 \times 10^5$ .

**Таблица 1.** Сравнение качества переупорядочения, полученного в конфигурациях (а) и (б) для библиотеки PMORSy. Обозначения: N – размерность матрицы, t\_c – тип огрубления, NZ – заполнение фактора матрицы.

матрица	N	t_c, (а)	NZ, (а)	t_c, (б)	NZ, (б)	NZ, ParMETIS
pwtk	217 918	HM	45 993 051	HM	47 284 604	47 124 530
msdoor	415 863	GPA(MWM)	52 749 023	GPA(MWM)	57 800 060	51 483 893
parabolic_fem	525 825	GPA(HM)	25 183 009	GPA(HM)	25 374 193	25 607 853
tmt_sym	726 713	HM	29 005 365	HM	29 856 527	29 507 621
boneS10	914 898	HM	323 277 648	GPA(HM)	313 408 788	268 565 124
Emilia_923	923 136	RM	1 700 731 236	LAM	1 733 187 171	1 636 886 316
audikw_1	943 695	HM	1 366 148 898	RM	1 359 524 322	1 216 865 448

**Таблица 2.** Сравнение времени работы библиотеки ParMETIS и PMORSy в конфигурации (б). Время работы в секундах. Обозначения: N – размерность матрицы, T\_1 – время работы в 1 поток, T\_8 – время работы в 8 потоков, t\_c – тип огрубления.

матрица	N	T_1, ParMETIS	T_8, ParMETIS	T_1, PMORSy	T_8, PMORSy	t_c
pwtk	217 918	1,77	1,26	0,63	0,21	HM
msdoor	415 863	2,37	2,03	1,00	0,32	RM
parabolic_fem	525 825	4,57	1,41	8,07	1,84	HM
tmt_sym	726 713	6,56	2,01	12,31	2,76	HM
boneS10	914 898	11,93	6,69	7,95	2,17	HM
Emilia_923	923 136	11,49	6,81	6,09	1,83	HM
audikw_1	943 695	17,72	10,98	9,06	2,68	HM

Проведенное исследование показало, что для задач, приоритетом которых является минимизация заполнения фактора, можно использовать алгоритм огрубления GPA со вспомогательной процедурой HM при выполнении небольшого числа шагов процедуры огрубления. Для задач, приоритетом которых является время работы переупорядочения и факторизации в целом, рационально использовать алгоритм HM. В дальнейшем в PMORSy будет встроена система автоматического выбора алгоритма переупорядочения по параметрам входной матрицы, а также выполнена интеграция алгоритма GPA в версию для систем с распределенной памятью.

## Литература

1. Pirova A., Meyerov I., Kozinov E., Lebedev S. PMORSy: parallel sparse matrix ordering software for fill-in minimization // Optimization Methods and Software. 2017. Vol. 32, No. 2. P. 274–289. DOI: 10.1080/10556788.2016.1193177
2. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs // SIAM J. on Scientific Computing. 1999. Vol. 20, No. 1. P. 359–392. DOI: 10.5555/305219.305248
3. Maue J., Sanders P. Engineering algorithms for approximate weighted matching // International Workshop on Experimental and Efficient Algorithms. Springer, Berlin, Heidelberg, 2007. P. 242–255. DOI: 10.1007/978-3-540-72845-0\_19
4. Monien B., Preis R., Diekmann R. Quality matching and local improvement for multilevel graph-partitioning // Parallel Computing. 2000. Vol. 26, No. 12. P. 1609–1634. DOI: 10.1016/S0167-8191(00)00049-1

5. LaSalle D. and Karypis G. Efficient Nested Dissection for Multicore Architectures // Euro-Par 2015: Parallel Processing. – 2015. – P. 467–478.

## Сравнение производительности вывода глубоких нейросетевых моделей при разных режимах запуска в Intel Distribution of OpenVINO toolkit\*

А.Е. Хватов, В.Д. Кустикова

Нижегородский государственный университет им. Н.И. Лобачевского

В настоящее время *глубокое обучение* (deep learning) применяется во многих прикладных областях. Существует большое количество нейросетевых моделей, обученных для решения классических задач *компьютерного зрения* (computer vision) и *обработки естественного языка* (natural language processing). Эти модели могут применяться напрямую, либо посредством *переноса обучения* (transfer learning), для решения сторонних прикладных задач. В этом случае особенно остро встает вопрос анализа производительности *вывода* (inference) моделей на доступных вычислительных ресурсах [1, 2]. Под выводом понимается однократный прямой проход по обученной сети с целью получения выхода модели для *пачки* (batch) примеров входных данных.

*Цель данной работы* – выполнить сравнительный анализ производительности вывода некоторых широко известных глубоких моделей, обеспечивающих решение задач *классификации изображений* (image classification), *детектирования объектов* (object detection) и *семантической сегментации изображений* (semantic segmentation), с использованием различных режимов исполнения, доступных в Intel Distribution of OpenVINO toolkit [3].

OpenVINO – набор инструментов для разработки эффективных приложений, основанных на применении глубокого обучения. В состав OpenVINO входит библиотека Inference Engine [3], предоставляющая программный интерфейс на C++ и Python для вывода нейронных сетей на аппаратных платформах Intel (Intel CPUs, Intel Processor Graphics, Intel Movidius VPU и др.). Inference Engine поддерживает два режима вывода: *режим минимизации времени выполнения одного запроса на вывод* (latency mode) и *режим минимизации времени выполнения набора запросов* (throughput mode). Первый режим предполагает одновременное выполнение только одного запроса на вывод (следующий запрос запускается по окончании предыдущего), второй – предусматривает возможность создания очереди запросов на вывод и их параллельное исполнение. Таким образом, если необходимо обработать набор запросов на вывод глубокой модели для набора пачек входных примеров, то в первом режиме реализуется их последовательная обработка, а во втором – параллельная обработка, при этом порядок исполнения запросов может быть произвольным, а порядок их старта/завершения не определен. Для реализации указанных режимов вывода предоставляется *синхронный* и *асинхронный программные интерфейсы*. Для вывода средствами Inference Engine модель конвертируется в *промежуточное представление*. Далее модель в промежуточном представлении загружается в плагин на исполнение. Доступно несколько плагинов для различных платформ Intel. Остановимся на плагинах, используемых в работе. Плагин CPU предназначен для вывода на Intel CPUs, плагин GPU – на Intel Processor Graphics, MYRIAD – на Intel Neural Compute Sticks. Также имеется *гетерогенный плагин* (heterogeneous plugin), который обеспечивает распределение обработки слоев сети между вычислительными устройствами Intel, и *плагин для нескольких устройств* (multi-device plugin), предоставляющий возможность выполнения запросов на вывод на разных устройствах.

Программная реализация вывода с использованием различных плагинов разработана в рамках системы измерения производительности вывода DLI [4]. Примерами аналогов системы являются DAWNBench [5] и MLPerf [6]. В процессе автоматического сбора результатов производительности разработанная система предоставляет результаты на трех аппаратных конфигурациях для более чем 20 публичных моделей и для ~30 моделей, обученных сотрудниками компании Intel [7]. Модели доступны в открытом репозитории Open Model Zoo [8]. В работе приводятся

---

\* Работа выполнена при поддержке компании Intel. Авторы благодарят сотрудников компании за помощь и внимание к работе.

результаты для избранных моделей, в частности, VGG-16 [9], SSD300 [10] и DeepLab-v3 [11]. Выбор моделей обусловлен принципиальными различиями архитектур сетей.

Анализ производительности выполняется на следующей тестовой инфраструктуре: Intel Core i7-8700 CPU @ 3.20GHz; 64 ГБ; Intel Gen9 HD Graphics; Intel Neural Compute Stick 2; Ubuntu 18.04.3 LTS; Python 3.7.3; Intel Distribution of OpenVINO toolkit 2020.2.

В качестве тестовых данных для проведения экспериментов используются выборки по 1024 изображения из открытых наборов ImageNet [12] и PASCAL VOC 2012 [13]. Выборка разбивается на отдельные пачки, при этом размер пачки является параметром эксперимента. Создается набор запросов на вывод. Вывод запускается средствами синхронного и асинхронного программных интерфейсов с использованием различных плагинов, описанных ранее. Для каждого эксперимента измеряется общее время обработки полной выборки при разном наборе параметров. Для гетерогенного плагина для каждой рассматриваемой модели выполняется подбор эффективного распределения слоев по устройствам. Для плагина для нескольких устройств выполняется подбор эффективного набора устройств и схемы расстановки им приоритетов.

Результаты экспериментов показывают эффективность применения сторонних устройств при необходимости разгрузки CPU. Применение гетерогенного плагина требует анализа времени выполнения вычислений на отдельных слоях модели на разных устройствах и подбора эффективной схемы распределения слоев между устройствами, что не является оправданным при наличии стандартных преобразований на слоях. Использование плагина для нескольких устройств позволяет минимизировать время выполнения набора запросов на вывод и имеет смысл при наличии потока пачек входных примеров (например, при обработке кадров видео).

## Литература

1. Jouppi N.P., et al. In-Datcenter Performance Analysis of a Tensor Processing Unit // Proceedings of the 44th Annual ISCA, 2017. P. 1–12. DOI: 10.1145/3079856.3080246.
2. Park J., et al. Deep Learning Inference in Facebook Data Centers: Characterization, Performance Optimizations and Hardware Implications. URL: <https://arxiv.org/abs/1811.09886> (дата обращения: 14.04.2020).
3. Intel Distribution of OpenVINO toolkit. URL: <https://software.intel.com/en-us/openvino-toolkit> (дата обращения: 14.04.2020).
4. Kustikova V., et al. DLI: Deep Learning Inference Benchmark // Communications in Computer and Information Science, 2019. Vol. 1129. P. 542–553. DOI: 10.1007/978-3-030-36592-9\_44.
5. DAWN Bench. An End-to-End Deep Learning Benchmark and Competition. URL: <https://dawn.cs.stanford.edu/benchmark> (дата обращения: 14.04.2020).
6. MLPerf [<https://mlperf.org>] (дата обращения: 14.04.2020).
7. DLI: Deep Learning Inference Benchmark. URL: <http://hpc-education.unn.ru/dli> (дата обращения: 14.04.2020).
8. OpenVINO Toolkit – Open Model Zoo repository. URL: [https://github.com/openai/open\\_model\\_zoo](https://github.com/openai/open_model_zoo) (дата обращения: 14.04.2020).
9. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. URL: <https://arxiv.org/pdf/1409.1556.pdf> (дата обращения: 14.04.2020).
10. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBoxDetector. URL: <https://arxiv.org/pdf/1512.02325.pdf> (дата обращения: 14.04.2020).
11. Chen L.-C., Papandreou G., Schroff F., Adam H. Rethinking Atrous Convolution for Semantic Image Segmentation. URL: <https://arxiv.org/pdf/1706.05587.pdf> (дата обращения: 14.04.2020).
12. ImageNet. URL: <http://www.image-net.org> (дата обращения: 14.04.2020).
13. PASCAL VOC 2012. URL: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012> (дата обращения: 14.04.2020).

## Сравнительный анализ 3D-структур белков на GPU: ре-имплементация алгоритма МАТТ на CUDA\*

И.А. Тимохин<sup>1</sup>, Д.А. Суплатов<sup>2</sup>, Н.Н. Попова<sup>1</sup>, В.К. Швядас<sup>2,3</sup>, Вл.В. Воеводин<sup>1,4</sup>

Факультет вычислительной математики и кибернетики<sup>1</sup>, Научно-исследовательский институт физико-химической биологии имени А.Н.Белозерского<sup>2</sup>, Факультет биоинженерии и биоинформатики<sup>3</sup>, Научно-исследовательский вычислительный центр<sup>4</sup>, Московский государственный университет имени М.В. Ломоносова

В процессе эволюции изменение аминокислотной последовательности происходит с большей скоростью, чем изменение 3D-структуры, что приводит к появлению групп родственных ферментов, обладающих аминокислотной идентичностью на грани статистической достоверности, но при этом характеризующихся существенной (70% и более) идентичностью по трехмерной структуре, а также принципам организации механизма функционирования. Сравнительный анализ 3D-структур белков, реализующих разнообразные функциональные свойства в рамках общей укладки суперсемейства, способен стать ключевым классом подходов биоинформатики к установлению взаимосвязи между структурой и функцией белков/ферментов, что важно как для фундаментального понимания организации и функционирования живых систем, так и для использования этих знаний в медицине и биотехнологии [1]. Однако в противоположность алгоритмам выравнивания аминокислотных последовательностей, которые основаны на поиске алфавитного сходства в эквивалентных позициях первичной структуры, целью пространственного выравнивания является геометрическое сравнение элементов третичной структуры, что является существенно более трудоемкой задачей с точки зрения объема вычислений. В настоящий момент в банке данных трехмерных структур белков PDB насчитывается более 162000 записей, число которых ежегодно растет. Эта тенденция прогрессирует и начинает опережать возможности ранее предложенных в биоинформатике программных решений задачи множественного 3D-структурного выравнивания. В этой связи, использование высокопроизводительных вычислений в биологии становится все более актуальным [2]. Ранее было продемонстрировано фундаментальное преимущество использования MPI/threads/OpenMP парадигм для сравнительного 3D-структурного анализа суперсемейств белков [3]. Целью этой работы является ре-имплементация популярного алгоритма МАТТ [3, 4] с поддержкой CUDA и оценка потенциала новой программы для решения задач биологии. Работа является пионерной, решения задачи множественного 3D-структурного выравнивания на GPU на настоящий момент отсутствуют.

Для переноса вычислений кода МАТТ на GPU потребовался детальный анализ как алгоритмов, реализующих отдельные этапы вычислений, так и организацию данных, используемых на различных этапах. Традиционный подход профилирования исходного кода МАТТ позволил выделить наиболее вычислительно-емкие функции алгоритма. В результате такого анализа были определены 3 основные функции с относительным временем вычислений в 36%, 13.7% и 13.3%. Время вычисления остальных функций было значительно меньше. Принципиальным моментом в переносе вычислений на GPU в выделенных функциях явилась организация структур данных. Используемая в этих функциях структура октодеревя не позволяла организовать эффективные вычисления на GPU. В связи с этим потребовалась реорганизация представления октодеревя и функций, работающих с этими структурами. Таким образом, перенос вычислений потребовал переработку алгоритма и его последующую верификацию, что является важным аспектом с точки зрения общего алгоритма структурного выравнивания. Рассмотренный аспект отличается от широко используемого подхода при переносе вычислений на GPU, заключающегося в CUDA-реализации «длинных» циклов.

---

\*\* Работа выполнена при поддержке гранта РФФИ № 20-07-00970 с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М.В. Ломоносова [5].

Первая версия новой программы CudaMATT была апробирована на наборах данных, которые соответствовали десяти самым большим по числу представителей суперсемействам белков по версии CATH, и содержали от 218 до 643 3D-структур со средним размером 91-348 аминокислотных остатка. Процедура подготовки выборки была описана ранее [3]. Для тестирования использовали следующие системы: (1) двухпроцессорный сервер на основе Intel Xeon E5-2630 v4 с nVidia GeForce RTX2080Super ASUS ROG-STRIX 8192Mb; (2) один узел раздела «pascal» суперкомпьютера «Ломоносов-2», содержащий Intel Xeon E5-2697 v3 и две Tesla P100, из которых использовалась только одна видеокарта; (3) вычислительный кластер ВМК МГУ, содержащий 5 узлов на основе двух процессоров POWER9 и двух графических ускорителей NVIDIA Tesla P100, из которых использовался один узел и один графический ускоритель. Предварительные результаты продемонстрировали существенные преимущества новой GPU-версии алгоритма MATT по сравнению с исходной CPU-версией [4]: использование современного GPU позволяет получить ускорение до 3 раз на отдельных выборках.

Текущие результаты убедили нас в перспективности выбранной стратегии использования GPU-ускорителей для решения задачи 3D-структурного выравнивания родственных белков, но также выявили некоторые проблемы. Дальнейшее развитие работы будет организовано по следующим направлениям: продолжение глубокой оптимизации структур и алгоритмов под архитектуру вычислений на графических ускорителях; поиск других частей алгоритма MATT, пригодных для переноса вычислений на GPU, и подготовка соответствующей реализации; адаптация полученного кода для использования более одного графического ускорителя на одном узле. На завершающем этапе работы планируется интеграция изменений, внесенных в MATT для реализации поддержки GPU, в ранее разработанную программу parMATT – версию MATT, поддерживающую вычисления на системах с распределенной памятью [3].

## Литература

1. Suplatov D., Sharapova Ya., Švedas V. Zebra2: advanced and easy-to-use web-server for bioinformatic analysis of subfamily-specific and conserved positions in diverse protein superfamilies // *Nucleic acids research*. – 2020. DOI:10.1093/nar/gkz385.
2. Suplatov D., Sharapova Y., Shegay M., Popova N., Fesko K., Voevodin VI., Švedas V. High-Performance Hybrid Computing for Bioinformatic Analysis of Protein Superfamilies // In *Russian Supercomputing Days 2019*. – Springer, Cham, 2019. – P. 249-264.
3. Shegay M.V. Suplatov D.A., Popova N.N., Švedas V.K., Voevodin VI.V. parMATT: parallel multiple alignment of protein 3D-structures with translations and twists for distributed-memory systems // *Bioinformatics*. – 2019. – V. 35. – No. 21. – С. 4456-4458.
4. Menke M., Berger B., Cowen L. Matt: local flexibility aids protein multiple structure alignment // *PLoS computational biology*. – 2008. – Т. 4. – №. 1. – P. e10.
5. Voevodin VI., Antonov A., Nikitenko D., Shvets P., Sobolev S., Sidorov I., Stefanov K., Voevodin Vad., Zhumatiy S. Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community // *Supercomputing Frontiers and Innovations*. – 2019. – V. 6. – No. 2. – P. 4-11



**Суперкомпьютерные дни в России**

**Аннотации полных статей на  
английском языке  
и приглашенных докладов**

### **3D model of wave impact on shore protection structures and algorithm of its parallel implementation (abstract)**

*Alexander Sukhinov, Aleksandr Chistyakov and Sofya Protsenko*

The present research considers the three-dimensional mathematical model of wave processes that allows to study the wave influence on shore protection structures and coastal infrastructure facilities. This model is based on Navier-Stokes equations of motion in areas with dynamically changing computational domain geometry. The pressure correction method is used to approximate the hydrodynamic model. The finite-difference schemes describing the mathematical model are constructed on the basis of integro-interpolation method using a scheme with weights. The adaptive variable-triangular iterative method is used to solve the system of grid equations. The numerical algorithms and program complex for their implementation practical significance is the possibility of their application in the hydrophysical processes study in the coastal water systems, as well as to build the water environment velocity and pressure fields and the hydrodynamic effects evaluation on shore protection structures and coastal structures.

Supported by RFBR, project 20-01-00421.

---

### **A Visual-Based Approach for Evaluating Global Optimization Methods (abstract)**

*Alexander Sysoyev, Maria Kocheganova, Victor Gergel and Evgeniy Kozinov*

In order to evaluate efficiency of some global optimization method or compare efficiency of different methods, it is necessary to select a set of test problems, define comparison measures, and, finally, choose a way of visual presentation of the computational results. In this paper, a wide set of test optimization problems is considered including a new global constrained optimization problem generator (GCGen). Main performance measures and comparative criteria of efficiency are presented. The ways of visual presentation of computational results are suggested.

Supported by RFBR, project 19-07-00242.

---

### **Adaptive global optimization using graphics accelerators (abstract)**

*Konstantin Barkalov, Ilya Lebedev and Vassili Toropov*

Problems of multidimensional multiextremal optimization and numerical methods for their solution are considered. The general assumption is made about the function being optimized: it satisfies the Lipschitz condition with an a priori unknown constant. Many approaches to solving problems of this class are based on reducing the dimension of the problem; i.e. addressing a multidimensional problem by solving a family of problems with lower dimension. In this work, an adaptive dimensionality reduction scheme is investigated, and its implementation using graphic accelerators is proposed. Numerical experiments on several hundred test problems were carried out, and they confirmed acceleration in the developed GPU version of the algorithm.

---

### **Analysis of Key Research Trends in High-Performance Computing using Topic Modeling Technique (abstract)**

*Yuri Zelenkov*

The intellectual structure of scientific discipline consists of a set of inter-acting topics. The evolution of these topics is the subject of special attention because it reflects the actual interest of researchers and stakeholders. This paper analyzes issues of High-Performance Computing

(HPC) on the base of the formal topic modeling technique. Analyzing the abstracts of 7661 publications referenced in Web of Science in 2005-2019, we identified seven topics that concern different aspects of HPC science. The central theme is the Large Scale Applications focused on practical and scientific problems solved using HPC. It is closely linked with Parallel Algorithms that should effectively exploit the thousands of processing cores, Parallel Software for heterogeneous distributed systems, and Interconnected systems that study the integration of HPC facilities in systems of larger size. These topics are relatively stable both in terms of popularity (number of publications) and impact (number of citations). The single topic, which popularity and impact continuously grow in the last 15 years, is Energy efficiency since power consumption is a critical issue of exascale systems. We also found that the topic of Heterogeneous systems dedicated mainly to GPU usage declines after the peak of interest in 2010-2015. The results obtained shed light on the structure of HPC science and supplement the known publications that declare research direction towards exascale performance.

---

**Application of a novel approach based on geodesic distance and pressure distribution to optimization of automated airframe assembly process (abstract)**

*Tatiana Pogarskaia, Maria Churilova and Elodie Bonhomme*

The paper is devoted to the parallel version of a new approach to temporary fastener pattern optimization. The main goal of the research is to improve the existing technological processes by reducing the number of installed temporary fasteners without compromising the assembly quality. Besides being combinatorial, the considered problem is complicated by the need to solve the contact problem on hundreds of input initial gaps for further analysis of the residual gap in the assembly after fastening. Commonly used heuristic methods become inapplicable to the regarded case due to their iterative based logic and the need for multiple calculations of the objective function value.

The proposed approach avoids iterative calculations and is based on calculation of the force distribution, needed to close an initial gap between joined parts, and geodesic distance between fasteners over the joint surfaces. The approach is applied and compared to a greedy algorithm previously used for the optimization of the robotized assembly process for an Airbus A350 S19 section.

---

**Application of supercomputing technologies for numerical implementation of an interaction graph model of natural and technogenic factors in shallow water productivity (abstract)**

*Alexander Sukhinov, Alla Nikitina, Alexander Chistyakov, Alena Filina and Vladimir Litvinov*

The paper covers the research and numerical implementation of an graph model of natural and technogenic factors' interaction in shallow water productivity. Based on it, the analysis of pulse propagation in computing environment from the vertices is performed in the context of research situation of valuable fish degradation of the Azov Sea that are subject to excessive commercial fishing withdrawal. The model takes into account the convective transport, microturbulent diffusion, taxis, catch, and the influence of spatial distribution of salinity, temperature and nutrients on changes in plankton and fish concentrations. Discrete analogue of proposed model problem of water ecology, included in software complex, were developed using schemes of second order of accuracy taking into account the partial filling of computational cells. The adaptive modified alternately triangular method was used for solving the system of grid equations of large dimension, arising at model discretization. Effective parallel algorithms were developed for numerical implementation of biological kinetics problem and oriented on multiprocessor computer system and NVIDIA Tesla K80 GPU with the data storage format

modification. Due to it, the production processes of biocenose populations of shallow water were analyzed in real and accelerated time.

Supported by RFBR, project 19-31-51017.

---

### **Availability-based Resources Allocation Algorithms in Distributed Computing (abstract)**

*Victor Toporkov and Dmitry Yemelyanov*

In this work, we introduce resources co-allocation algorithms for parallel jobs execution in distributed computing with non-dedicated and heterogeneous hosts. Complex distributed computing systems often operate under conditions of the resources availability uncertainty. Imprecise estimations of jobs execution runtime, unplanned maintenance works and other global and local events do not allow to consider accurate schedules of the resources utilization. On the other hand, an efficient job-flow execution in compliance with QoS constraints requires reliable mechanisms for advanced resources allocation and reservation. The novelty of the proposed resources allocation approach is in a general procedure efficiently selecting computing nodes according to the resources availability criteria. Special knapsack and greedy algorithms are implemented and compared in a market-based computing simulation model.

Supported by RFBR, projects 18-07-00456, 18-07-00534.

---

### **Core algorithms of sparse 3D mipmapping visualization technology (abstract)**

*Stepan Orlov, Alexey Kuzin and Alexey Zhuravlev*

The paper presents algorithms implemented in the core components of sparse 3D mipmapping technology, which is a basement for animated visualization of scalar time-dependent fields defined on large (up to  $1e10$  nodes) meshes and resulting from CFD simulations. The basic idea of sparse 3D mipmapping is the interpolation of original data on octree meshes and further visualization of fields on subtrees of limited depth using volume rendering algorithms. The paper focuses on data structures and algorithms delivering input for visualization, and presents the results of performance testing.

---

### **Describing HPC System Architecture for Understanding its Capabilities (abstract)**

*Dmitry Nikitenko, Alexander Antonov, Artem Zheltkov and Vladimir Voevodin*

There is a variety of known HPC ratings nowadays which represent machine capability for solving a fixed problem, based on a certain algorithm, but these ratings represent a top of the iceberg, and as a rule, one can't compare application tuning features even for the selected system, and the details of system architecture are not usually described precisely. At the same time lots of efforts are made to describe diverse algorithm features formally, AlgoWiki is one of the most notable recent projects. The idea of Algo500 is joining precise description of computer system with detailed formal descriptions of algorithms using implementation performance data, and building an engine over such joint base to allow various queries, thus giving means of building user-defined ratings regarding selected method, algorithms and/or computer platform features. This paper gives an overview of Algo500 design and some use cases.

---

### **Developing efficient implementation of label propagation algorithm for modern NVIDIA GPUs (abstract)**

*Ilya Afanasyev and Dmitry Lichmanov*

Systems equipped with modern NVIDIA GPUs nowadays are capable of solving many real-world problems, including large-scale graph processing. Efficiently implementing graph algorithms on GPUs is a challenging task, since modern real-world graphs have irregular structure. Current paper describes approaches, which can be used for developing efficient implementation of label propagation algorithm, frequently used to solve graph clustering and community detection problems. Compared to already existing GPU-based implementations, new optimization techniques have been proposed, including graph preprocessing, efficient load-balancing, using unified memory for out-of-core graph processing, and several others. The performance of the developed implementation has been evaluated on synthetic and medium-scaled real-world graphs, resulting in up to 2 times better results compared to existing approaches.

Supported by RFBR, project 19-31-27001.

---

**Development of experimental data processing workflows based on Kubernetes infrastructure and REANA workflow management system (abstract)**

*Anton Teslyuk, Sergey Bobkov, Alexey Poyda, Alexander Novikov, Vasily Velikhov and Viacheslav Ilyin*

In this paper we present the design of data processing workflow for scientific experiments, which require complicated multi-step analysis procedure. We test it on datasets from Single Particle Imaging (SPI) experiments. The workflow is based on microservice architecture, Docker containers and Kubernetes platform. For workflow setup and management we use REANA software which is compatible with Kubernetes orchestrator and supports standard Common Workflow Language (CWL) to describe complex computing jobs. Our approach allows easy construction of workflows of diverse architecture for a wide range of applications. It allows integration of heterogeneous software in a uniform way as well as easy modification or replacement of workflow components. In the same time it allows easy scaling of computations in a cloud infrastructure. We show the applicability of the designed scheme and estimate the overhead of the platform middleware.

Supported by RFBR, project 18-29-23020.

---

**Different partitioning algorithms study applied to a problem of digital rock physics (abstract)**

*Evdokia N. Golovchenko, Mikhail V. Yakobovskiy, Vladislav A. Balashov and Evgeny B. Savenkov*

In recent years digital rock physics technology is regarded as a promising tool which can supplement traditional laboratory techniques. This technology is based on numerical experiment with direct resolution of pore space of a rock sample, which is obtained with computed microtomography. Necessity of high resolution leads to a high dimension of discrete settings ( $10^6$  -  $10^9$  numerical cells). The work is devoted to an application of different partitioning algorithms to the problem of flow simulation within geometry of rock sample pore space. Simulation of a single-phase fluid flow within pore space of a sandstone sample with voxel representation is used to compare the partitions obtained by various methods using parallel partitioning tools ParMETIS, Zoltan, and GridSpiderPar. Average time spent on interprocess exchange during 200 time steps of the considered parallel simulation was compared when the grid was distributed over the cores in accordance with various partitions. The obtained results demonstrate advantages of some algorithms and reveal the criteria, important for the problem.

The study is conducted with support from The Ministry of Education and Science of Russian Federation, unique identifier of the Project RFMEFI60419X0209

---

### **Discrete Event Simulation Model of a Desktop Grid System (abstract)**

*Evgeny Ivashko, Natalia Nikitina and Alexander Rumyantsev*

The paper describes a discrete event simulation model of a Desktop Grid system. Firstly, we present a stochastic model of a volunteer computing project. We then employ the event simulation approach based on the generalized semi-Markov processes to develop a discrete event simulation model. Finally, using the simulation model, we describe a performance optimization problem aiming to optimize the project runtime as a function of the task size under performance constraints.

Supported by RFBR, projects 18-07-00628, 19-57-45022, 19-07-00303, 18-07-00156, 18-07-00147.

---

### **Drop oscillation modeling (abstract)**

*Lev Shchur and Maria Guskova*

The classical problem of oscillations of liquid droplets is a good test for the applicability of computer simulation. We discuss the details of our approach to a simulation scheme based on the Boltzmann lattice equation. We show the results of modeling induced vibrations in a chain of three drops in a closed tube. In the initial position, the central drop has formed as an ellipsoid, out of the spherical equilibrium form. The excitation of vibrations in the left and right droplets depends on the viscosity of the surrounding fluid and the surface tension. Droplets are moving out of the initial position as well. We discuss the limits of the applicability of our model for the study of such a problem. We will also show the dynamics of the simulated process.

The work is carried out according to the project of the Russian Science Foundation 19-11-00286 and partially according to the RFBR project 20-07-00145.

Supported by RFBR, project 20-07-00145.

---

### **Enumerating the orthogonal diagonal Latin squares of small order for different types of orthogonality (abstract)**

*Eduard Vatutin and Alexey Belyshev*

The article describes computational experiments aimed to enumerating the number of orthogonal diagonal Latin squares for general and special types of orthogonality. General type orthogonality can be verified using Euler-Parker method, corresponding number of main classes of orthogonal diagonal Latin squares, number of normalized orthogonal diagonal Latin squares and total number of orthogonal diagonal Latin squares of general type form previously unknown numerical series A330391, A305570 and A305571 (calculated up to order 8) added to OEIS. Self-orthogonal (SODLS), doubly self-orthogonal (DSODLS) and extended self-orthogonal diagonal Latin squares (ESODLS) are form a set of special types of orthogonality. For each of these types corresponding numerical series was calculated and published in OEIS with numbers A329685, A287761, A287762 (SODLS, up to order 10), A333366, A333367, A333671 (DSODLS, up to order 10) and A309210, A309598, A309599 (ESODLS, up to order 8). Values for orders 1-8 were obtained by analyzing the complete lists of canonical forms of the main classes of orthogonal DLSs obtained by the authors by exhaustive search. Values for order 9 were derived from the SODLS list of order 9 provided by Harry White. Values for order 10 were obtained by analyzing the list of SODLS of order 10, available online (van Vuuren et

al.). The values obtained confirm the similar values for SODLS and DSODLS obtained previously by Francis Gasparou and partially published by Harry White. For some of the obtained numerical values previously unknown mathematical relations are empirically established:  $A_{287761}(n) \cdot 4 = A_{309598}(n)$ ,  $A_{287762}(n) \cdot 4 = A_{309599}(n)$ , for  $n \geq 7$ ;  $8g$ ,  $A_{333366}(10) = A_{333367}(10) = A_{333671}(10) = 0$ ,  $A_{309210}(10) = 33-240$ .

---

### **High-Performance Simulation of high-beta plasmas using PIC method (abstract)**

*Igor Chernykh, Vitaly Vshivkov, Galina Dudnikova, Tatyana Liseykina, Ekaterina Genrikh, Anna Efimova, Igor Kulikov, Ivan Chernoshtanov and Marina Boronina*

In this paper, we present the new parallel PIC code for numerical simulation of plasma physics problems. We used previously developed \cite{Dudnikova} hybrid model of plasma with new realisation of numerical method which is adapted for vectorization for modern CPUs. We tested our code on a very important high-energy physics problem which is connected with thermonuclear synthesis. Numerical simulation of different types of diamagnetic traps can be used for prototyping of magnetic trap devices for compact fusion reactors. As a result of this work, we proposing some parallel tests of our code and numerical simulation of diamagnetic “bubble”.

---

### **Implementation of SL-AV global atmosphere model with 10km resolution (abstract)**

*Mikhail Tolstykh, Gordey Goyman, Rostislav Fadeev and Vladimir Shashkin*

Huge computer resources needed to compute the 24-hours global weather forecast in less than 20min dictate the necessity to optimize the numerical algorithms of the model and their parallel implementation. We present some experience gained while implementing the new high-resolution version of the SL-AV global atmosphere model for numerical weather prediction at parallel systems with many thousands of processor cores. Unlike our previous scalability studies, we need to minimize the elapsed time of the forecast at given processor cores number which is currently about 4000. The results of optimizations are shown for two Roshydromet computer systems, one of them is Cray XC40.

Supported by RFBR, project 19-31-90032.

---

### **INMOST platform for parallel multi-physics applications: multi-phase flow in porous media and blood flow coagulation (abstract)**

*Kirill Terekhov, Kirill Nikitin and Yuri Vassilevski*

INMOST (Integrated Numerical Modeling Object-oriented Supercomputing Technologies) is an open-source platform for fast development of efficient and flexible parallel multi-physics models. In this paper we review capabilities of the platform and present two INMOST-based applications for parallel simulations of multi-phase flow in porous media and clot formation in blood flow.

The finite volume (FV) method is the popular approach to spatial discretizations on general meshes (i.e. meshes composed of general polyhedral cells), especially for geophysical and biomedical applications where local mass conservation is vital. INMOST provides a complete set of tools for development of FV discretizations for linear and nonlinear problems: automatic differentiation tool for assembly of the nonlinear residual and corresponding Jacobian and Hessian matrices, iterative solvers of nonlinear systems arising from PDEs discretization, parallel solvers for sparse linear algebraic systems.

The platform also provides a technology for development of numerical models on general unstructured grids. It includes parallel mesh data structures, low-level infrastructure for reading, writing, creating, manipulating and partitioning of distributed general meshes.

The synergy of INMOST platform and efficient FV discretizations for systems of PDEs on general meshes produces a powerful tool for supercomputing simulations.

Supported by RFBR, project 18-31-20048.

---

### **Kirchhoff-type implementation of multi-arrival 3-D seismic depth migration with amplitudes preserved (abstract)**

*Alexandr Pleshkevich, Anton Ivanov, Vadim Levchenko and Sergey Khilkov*

We present implemented algorithm of asymptotic 3d prestack seismic migration, that takes into account multi-arrival and caustics of ray Green's functions (GF). Seismic migration is a crucial stage in processing of the seismic prospecting data. It requires dozens of thousands and in some cases hundreds of thousands of hours of processor time. The seismic migration problem naturally decomposes into two parts. The first one is computation of the GFs for all sources and the second one is summation. In order to compute a GF with multipath ray propagation taken into account one has to use adaptive grids to approximate the wave front. It is also necessary to implement a special "capture" procedure that calculates parameters at points of the uniform grid. This procedure also deals with singularities arising on caustic surfaces. An efficient implementation of the summation part requires to move lots of GF fragments between cluster nodes. We have employed described algorithms in a program and tested it on synthetic datasets and a few real world projects.

This research was initiated and sponsored by Central Geophysical Expedition JSC of Rosgeo.

---

### **LLVM based parallelization of C programs for GPU (abstract)**

*Nikita Kataev*

The paper proposes an approach to semi-automatic program parallelization in SAPFOR (System FOR Automated Parallelization). SAPFOR proposes opportunities to perform user-guided source-to-source program transformations and to reveal implicit parallelism in sequential programs. The LLVM compiler infrastructure is used to examine a program and Clang is used to perform source-to-source program transformation. This paper highlights benefits of IR-level program analysis which allows us to apply low-level program transformations to investigate properties of the original program. To exploit program parallelism SAPFOR relies on DVMH which is a directive-based programming model. We use subset of C-DVMH language which allows us to run parallel program on GPU as well on multiprocessors. Evaluation of presented approach has been performed using the C version of the NAS Parallel Benchmarks.

---

### **Management of computations with LRnLA algorithms in adaptive mesh refinement codes (abstract)**

*Anton Ivanov, Vadim Levchenko, Boris Korneev and Anastasia Perepelkina*

The data structure for codes with adaptive mesh refinement with low overhead for data storage is developed. The computational fluid dynamics scheme is implemented on this structure with the use of the ConeFold locally recursive non-locally asynchronous algorithm. It increases the computational intensity by a recursive traversal of the parts of the dependency graph in a ConeFold. The dependencies are determined by the mesh structure, and the mesh may be

coarsened or refined at the synchronization time moments. For an efficient execution in the parallel environment of the many-core and many-node systems the ConeFold task manager is used. The load balancing is performed with an account for the computation complexity of different ConeFolds. The work is supported by the Russian Science Foundation, grant # 18-71-10004.

---

**Mathematical Modeling of Sustainable Coastal Systems Development Scenarios Based on Game-Theoretic Concepts of Hierarchical Management Using Supercomputer Technologies (abstract)**

*Yulia Belova, Alexander Chistyakov, Alla Nikitina and Vladimir Litvinov*

The work is devoted to mathematical modeling of the scenario for managing the sustainable development of coastal systems using the example of the Azov Sea. The dynamic problem of minimizing the costs of maintaining the ecosystem of the reservoir in a given state, which is interpreted as a requirement for sustainable development, is being solved. The mathematical model of the interaction of two phytoplankton types takes into account the influence of abiotic factors, such as salinity and temperature, on the algae growth, their absorption and excretion of nutrients, as well as the transition of these nutrients from one form to another. For the numerical implementation of the proposed interconnected mathematical models of biological kinetics, parallel algorithms have been developed that are adapted to multiprocessor and hybrid computing systems using the NVIDIA CUDA architecture. An analysis of the characteristics of the CUDA architecture showed the applicability of the algorithms for the numerical implementation of the developed mathematical models of hydrobiology to create high-performance information systems. The constructed software package allows to simulate the problem of reducing eutrophication and toxicity of the coastal system under consideration by displacing harmful blue-green algae with a cultivated strain of green alga *Chlorella vulgaris* BIN.

Supported by RFBR, project 20-01-00421.

---

**Multiple-Precision BLAS Library for Graphics Processing Units (abstract)**

*Konstantin Isupov and Vladimir Knyazkov*

The binary32 and binary64 floating-point formats provide good performance on current hardware, but also introduce a rounding error in almost every arithmetic operation. Consequently, the accumulation of rounding errors in large computations can cause accuracy issues. One way to prevent these issues is to use multiple-precision floating-point arithmetic. This paper presents a new library of basic linear algebra operations with multiple precision for graphics processing units. The library is written in CUDA C/C++ and uses the residue number system to represent multiple-precision significands of floating-point numbers. The supported data types, memory layout, and main features of the library are considered. Experimental results are presented showing the performance of the library.

---

**New compact streaming in LBM with ConeFold LRnLA algorithms (abstract)**

*Anastasia Perepelkina, Vadim Levchenko and Andrey Zakirov*

The classic Lattice Boltzmann Method with a cube-shaped stencil is a memory-bound problem, and the optimization techniques aim for a goal of one load and one store per value update. We propose a data layout and a streaming pattern so that the elementary computation is an update of 8 cells in a cube, for which only the data of the same cells are required. The new streaming is symmetrical, compact, and allows split description.

With it, the better localization of computation in the higher levels of the CPU memory hierarchy is possible, which leads to the use of the higher ceilings in the Roofline model. It is implemented with the ConeFold LRnLA algorithm so that the operational intensity is increased further. The compactness of the streaming allows for more asynchrony in multidimensional LRnLA decomposition. The obtained one desktop CPU performance reaches 1 billion cell updates per second.

---

**Nonlinear bending instabilities accompanying clump and filament formation in collisions of nebulae (abstract)**

*Valery Goryachev and Boris Rybakin*

It is known that the nucleation of stars in the universe occurs, as a rule, in areas of the interstellar medium, where nebulae - gas and dust clouds consisting of molecular hydrogen, can collide with each other, changing their state during dynamic interaction with gravitational and magnetic fields. The gravitational-turbulent description of these processes is quite common in explaining the reasons for the creation of pre-stellar regions as a consequence of the collision of molecular clouds. We adhere to this approach with some simplification, assuming that the main factor in the dynamic transformations of gas formations is the influence on the collision process of mainly the kinetic energy of the clouds, separating these effects from the effects of gravitational collapse and from the influence of magnetic fields. To simulate gas-dynamic processes of different scales, a parallel numerical code was developed using grids with improved resolution, which was used in a numerical experiment on high-performance computers. The simulation showed that sharp changes in the distribution of matter in the shock core of new formations can be triggered by the Kelvin-Helmholtz instability and nonlinear thin-shell instability, which lead to sharp perturbations of the gas density in the resulting clumps, outer shells and gas filaments, with density fluctuations in the external interstellar medium. A predictive analysis of the appearance of possible pre-stellar zones during the evolution of new cloud formations is given

Supported by RFBR, project 19-29-09070.

---

**Numerical Forecast of Local Meteorological Conditions on a Supercomputer (abstract)**

*Alexander Starchenko, Sergey Prohanov, Eugeny Danilkin and Dmitry Lechinsky*

A high-resolution mesoscale meteorological model for forecasting and studying weather events and surface air quality in an urbanized area or a large industrial or transportation hub is presented.

An semi-implicit second-order finite volume method with parallel implementation on multiprocessor computing system was developed to solve the equations of the model. The results of testing the parallel program on the supercomputer Cyberia of Tomsk State University demonstrated its high efficiency. The approach developed was successfully applied to predicting heavy precipitation events and an urban heat island effect.

---

**Optimization of load balancing algorithms in parallel modeling of objects using a large number of grids (abstract)**

*Vladislav Fofanov and Nikolay Khokhlov*

This work aims to find the optimal algorithm of computational load distribution for parallel modeling of objects using multiple grids. Reducing of calculation time will be achieved by reducing the amount of information transmitted between processes through contact surfaces.

For testing effectiveness of the algorithms, an elastic wave propagation model in a medium with many non-parallel cracks will be used.

Modeling with using of researched algorithms was carried out on the “Kurchatov Data Processing Center” programming complex, which is part of the “Kurchatov Institute” National Research Center.

Supported by RFBR, project 18-31-20041.

---

**Parallel BIILU2-based iterative solution of linear systems in reservoir simulation: do optimal parameters exist? (abstract)**

*Igor Konshin, Kirill Nikitin, Kirill Terekhov and Yuri Vassilevski*

Three-phase black oil model for SPE-10 test problem with highly heterogeneous permeability and porosity fields was considered.

A set of linear systems by the research parallel three-phase fully implicit black oil simulator was generated.

The parallel performance of the MPI-based BIILU2 linear solver with the default parameters on such systems was analyzed.

The trends in the choice of optimal BIILU2-based linear solver parameters for such systems was determined.

---

**Parallel Box-Counting Method for Evaluating the Fractal Dimension of Analytically Defined Curves (abstract)**

*Ilya Pershin, Dmitrii Tumakov and Angelina Markina*

Serial and parallel implementations of the algorithm intended for calculating the number of boxes for evaluating the fractal dimension of analytical curves are considered. The algorithm contains four stages: 1) preparing the data for calculations; 2) determining the boxes into which the curve fell; 3) counting the boxes that have an intersection with a curve; 4) counting the boxes of a larger size that intersect with a curve. The acceleration of computations performed by a parallel code (on OpenMP and CUDA) with respect to calculations performed by a serial code depending on the size of the box is investigated. Numerical experiments are carried out, the results of which exhibit a significant increase in performance for GPU calculations in the case of a large number of segments of the curve. A 100-fold increase in the computational speed is obtained for a curve containing a million segments with a billion boxes (box size is  $2^{(-15)}$ ). The graphs depicting an increase in acceleration of parallel code performance with decreasing the box size and increasing the number of curve segments are shown. It is concluded that the efficiency of using the GPU begins with three million boxes and grows with an increase in the number of curve points.

Supported by RFBR, project 19-31-90129.

---

**Parallel efficiency of time-integration strategies for the next generation global weather prediction model (abstract)**

*Vladimir Shashkin and Gordey Goyman*

Next generation global numerical weather prediction models will have horizontal resolution of 3-5 km that lead to the problem of about  $10^{10}$  degrees of freedom. To meet operational requirements for medium-range weather forecast,  $O(10^4)$ - $O(10^5)$  processor cores have to be

used efficiently. The non-hydrostatic equation set will be used so models have to treat efficiently a number of fast-propagating wave families. Therefore, time-integration scheme is crucial for the scalability and computational efficiency.

The next generation global atmospheric model is currently under development at INM RAS and Hydrometcentre of Russia. To choose the time-integration scheme for the new model, we evaluate scalability and efficiency of several options (including exponential propagation integrator) using linearized equation set. We also present a parallel framework developed for the solution of atmospheric dynamic equations on the spherical cube grid.

---

### **Parallel Gravitational Search Algorithm in Solving the Inverse Problem of Chemical Kinetics (abstract)**

*Leniza Enikeeva, Dmitry Smirnov, Irek Gubaydullin and Mikhail Marchenko*

The article describes a parallel gravitational search algorithm and its application to solving the inverse problem of chemical kinetics. The relevance of the study of metaheuristic algorithms, including the gravitational search algorithm, is given. It is shown that recently, these algorithms are becoming increasingly popular. The optimization problem is formulated on the example of solving the inverse kinetic problem. The process under study is propane pre-reforming over Ni catalyst, which is an industrially important chemical process. The description of the algorithm and its pseudocode are presented, after which the performance of the gravitational search algorithm is compared with other metaheuristic methods. The algorithm demonstrated its competitiveness, as a result of which it was applied to solve a specific industrial problem. Using this algorithm, the direct and inverse problems of chemical kinetics are solved, and the optimal values of the kinetic parameters of the reaction are found. It is proved that the model correctly describes the available experimental data.

Supported by RFBR, projects 19-37-60014, 18-01-00599.

---

### **Parallel Multilevel Linear Solvers within INMOST Platform (abstract)**

*Kirill Terekhov*

The work is dedicated to domain-decomposition parallel iterative linear system solver. An algebraic multilevel preconditioner is used on a subdomain. The cornerstone of the method is the dual-threshold second-order Crout incomplete LU factorization. The Crout version of LU factorization permits condition estimation for inverse factors. The factorization of the current row and column of the matrix is postponed upon the growth of the estimated condition. Following factorization, the Schur complement is computed for the postponed part and the factorization continues in the multilevel fashion until the complete matrix is factorized. Before each level factorization begins, the matrix is first permuted by finding maximum transversal and reverse Cuthill-McKee permutations and then rescaled into I-dominant matrix. The performance of the method on a coupled multiphysics problem of blood coagulation is demonstrated.

---

### **Predictive quantum-chemical design of molecules of high-energy heterocyclic compounds (abstract)**

*Vadim Volokhov, Tatyana Zyubina, Alexander Volokhov, Elena Amosova, Dmitry Varlamov and David Lempert*

Rapidly developing new technologies, especially in the field of modern aircraft, stimulate great interest in creating high-energy materials for various purposes. Recently, modern computer technologies have been playing an increasingly important role in creating new materials with

determined properties. Focus of this work is on computer design of new compounds that have not yet been synthesized – high-enthalpy derivatives of heterocycles, such as tetrazine, furazan, furoxan, triazole, etc. using quantum chemical supercomputer calculation methods. For experimentally studied substances C<sub>2</sub>N<sub>6</sub>O<sub>4</sub>, C<sub>2</sub>N<sub>6</sub>O<sub>3</sub>, C<sub>2</sub>N<sub>8</sub>O<sub>4</sub>, the calculated values of enthalpy  $\Delta_f H_{298}$  (g) are 8-15% higher than the experimental values, which is significantly less than the spread of experimental values for these compounds. The enthalpy of formation of the studied gaseous molecules was calculated using the atomization method. The simulation was performed within the GAUSSIAN 09 software package using the B3LYP hybrid density functional with the basis 6-311+G(2d,p) and the combined methods G4 and G4(MP2). Tasks have high computational complexity and the calculation time it takes from several hours to months on multi-node supercomputer configurations.

Supported by RFBR, project 20-07-00319.

---

### **Privacy-Preserving Logistic Regression as a Cloud Service based on Residue Number System (abstract)**

*Jorge Mario Cortés-Mendoza, Andrei Tchernykh, Mikhail Babenko, Luis Bernardo Pulido-Gaytán, Gleb Radchenko, Franck Leprevost, Xinheng Wang and Arutyun Avetisyan*

The security of data storage, transmission, and processing is emerging as an important consideration in many data analytics techniques and technologies. For instance, in machine learning, the datasets could contain sensitive information that cannot be protected by traditional encryption approaches. Homo-morphic encryption schemes and secure multi-party computation are considered as a solution for privacy protection. In this paper, we propose a homomorphic Logistic Regression based on Residue Number System (LR-RNS) that provides security, parallel processing, scalability, error detection, and correction. We verify it using six known datasets from medicine (diabetes, cancer, drugs, etc.) and genomics. We provide experimental analysis with 30 configurations for each dataset to compare the performance and quality of our solution with the state of the art algorithms. For a fair comparison, we use the same 5-fold cross-validation technique. The results show that LR-RNS demonstrates similar accuracy and performance of the classification algorithm at various thresholds settings but with the reduction of training time from 85.4% to 97.5%.

Supported by RFBR, project 18-07-01224.

---

### **Quantum Software Engineering: Quantum gate-based computational intelligence supremacy (abstract)**

*Olga Ivancova, Vladimir Korenkov, Nikita Ryabov and Sergey Ulyanov*

A new approach to a circuit implementation design of quantum algorithmic gates for quantum massive parallel computing realization is presented. The main attention is focused on the development of design method of fast quantum algorithm operators as superposition, entanglement and interference which are in general time-consuming operations due to the number of products that have to be performed. SW&HW support toolkit of supercomputing accelerator of quantum algorithm simulation is described. The method for performing Grover's interference without product operations introduced. New quantum genetic and quantum fuzzy inference algorithm gate design considered.

Supported by RFBR, projects 18-02-40101, 18-07-01359.

---

### **Replication of “tail” computations in a Desktop Grid project (abstract)**

*Evgeny Ivashko and Natalia Nikitina*

The paper addresses the problem of accelerating the “tail” stage of a computational experiment in a Desktop Grid. We provide the mathematical model of a “tail” stage, describe the setting of simulation experiments and provide their results. Task replication in “tail” phase proves to be efficient in decreasing the duration of “tail” by orders of magnitude.

Supported by RFBR, project 18-07-00628.

---

**Research of hardware implementations efficiency of sorting algorithms created by using Xilinx’s High-Level Synthesis tool (abstract)**

*Alexander Antonov, Denis Besedin and Alexey Filippov*

The article describes results of our research of hardware implementation efficiency of sorting algorithms created by using of Xilinx's High-Level Synthesis tools, the Vivado HLS package, and FPGAs. The term efficiency, used in the research, defined as a function of the performance, estimated in time for sorting a random generated array, and hardware “cost”, estimated in utilized FPGA re-sources. In the research, a simulation modeling and a comparative analysis was carried out for the wide range of sorting algorithms implemented on a universal processor and on the Xilinx’s FPGAs. The research results prove that hardware implementations of the sorting algorithms, synthesized by Xilinx HLS tool, do not always provide higher performance comparing with the implementations of the same algorithms on a universal processor. The article shows that the hardware implementation of the Merge sort algorithm, created by Xilinx’s HLS tool, can speed up, comparing with software implementation, the process of sorting arrays of small and medium size.

Supported by RFBR, project 18-29-03250.

---

**Resource-Efficient Parallel CG Algorithms for Linear Systems Solving on Heterogeneous Platforms (abstract)**

*Nikita S. Nedozhogin, Sergey P. Kopysov and Alexandr K. Novikov*

The article discusses the parallel implementation of solving systems of linear algebraic equations on computational nodes containing a central processing unit (CPU) and graphic accelerators (GPU). The performance of parallel algorithms for the classical conjugate gradient method schemes when using the CPU and GPU together is significantly limited by the synchronization points. The article investigates the pipeline version of the conjugate gradient method with one synchronization point, the possibility of asynchronous calculations, load balancing between the CPU and GPU when solving the large linear systems.

Numerical experiments were carried out on test matrices and computational nodes of different performance of a heterogeneous platform, which allowed us to estimate the contribution of communication costs. The algorithms are implemented with the joint use of technologies: MPI, OpenMP and CUDA.

The proposed algorithms, in addition to reducing the execution time, allow solving large linear systems, for which there are not enough memory resources of one GPU or a computing node.

At the same time, block algorithm with the pipelining decreases the total execution time by reducing synchronization points and aggregating some messages in one.

Supported by RFBR, project 17-01-00402.

---

**Risky search with increasing complexity by a Desktop Grid (abstract)**

*Ilya Chernov and Evgeny Ivashko*

A common problem solved by high-performance computing is a search problem when the unique object needs to be found among other objects. With a huge number of objects to examine and a computationally hard examination of each, the search problem requires a lot of computing resources. However, the problem becomes even harder if an examination might give the wrong results with some probability. Such problem appears in unreliable high-throughput computing environments like Desktop Grids. In this paper, we present a mathematical model of such search problems, derive the optimal strategy of task assignment that minimizes the expected cost of examinations and thus reduces the consumption of computing resources. We show that in a rather general case the optimal strategy is the "no-replication" one, i.e., all objects should be examined once, then for the second time if no target has been obtained, etc. We reveal the cases when this strategy is not optimal, e.g., when the cost of examination for objects grows too quickly, or when the cost of re-examination of the same object decreases too quickly. Also, the expected costs of finding the target are obtained for a practical case of object-dependent cost.

Supported by RFBR, project 18-07-00628.

---

### **Running Many-Task Applications Across Multiple Resources with Everest Platform (abstract)**

*Oleg Sukhoroslov, Vladimir Voloshinov and Sergey Smirnov*

Distributed computing systems are widely used for the execution of loosely coupled many-task applications, such as parameter sweeps, workflows, distributed optimization. These applications consist of a potentially large number of computational tasks that can be executed more or less independently. Since the application users often have an access to multiple computing resources, it is important to provide a convenient and efficient environment for execution of applications across the user-defined heterogeneous resource pools. The paper discusses the related challenges and presents an approach for solving them based on Everest, a web-based distributed computing platform. The presented solution supports reliable and efficient execution of many-task applications, while taking into account resource performance, adapting to queuing delays and providing a mechanism for communication between tasks.

Supported by RFBR, projects 18-07-00956, 20-07-00701.

---

### **Set Classification in Set@I Language for Architecture-Independent Programming of High-Performance Computer Systems (abstract)**

*Ilya Levin, Alexey Dordopulo, Ivan Pisarenko and Andrey Melnikov*

Traditional programming languages for parallel computer systems do not separate the description of an algorithm from the details of its hardware implementation efficiently. As a result, the porting of the same algorithm between different computational architectures and configurations requires a significant code modification. To reduce the time and complexity of porting, we proposed an architecture-independent Set@I programming language based on the aspect-oriented programming paradigm and set-theoretical code view. In contrast to conventional tools for parallel programming, Set@I operates by sets, subsets, attributes and relations between them. Various aspects of realization transform an architecture-independent source code according to the certain architecture and configuration of a computer system. Set@I provides the porting of parallel applications without the changing of a source code, the modification of an information graph or a computing structure with regard to computer system's features or user's preferences, and the indefinite description of calculations. This paper treats the essential issue of set classification in the Set@I programming language. We propose three basic criteria of set typing: by the parallelism of collection's elements, by the definiteness, and

by the processing method. We yield examples of programs in Set@1 that demonstrate the techniques and specificities of various set types' usage.

Supported by RFBR, project 20-07-00545.

---

### **Shared Memory based MPI Broadcast Algorithms for NUMA systems (abstract)**

*Mikhail Kurnosov and Elizaveta Tokmasheva*

MPI\_Bcast collective communication operation is used by many scientific applications and tend to limit overall parallel application scalability. As the number of cores per computer node keeps increasing, it becomes important for MPI to leverage shared memory for intranode communication. This paper investigates the design and optimization of broadcast operation for NUMA nodes with GNU/Linux. We describe algorithms for MPI\_Bcast that take advantage of NUMA-specific placement of queues in a shared memory for message transferring. On a Xeon Nehalem and Xeon Broadwell servers, our implementation achieves on average 20–60% speedup over algorithms of Open MPI coll/sm and MVAPICH.

Supported by RFBR, project 18-07-00624.

---

### **Similarity Mining of Message Passing Delays in Supercomputer Networks Based on Peak and Step Detection (abstract)**

*Artur Begaev, Alexey Salnikov and Archil Maysuradze*

The problem of analyzing the delays arising in the transmission of data in a multiprocessor computing system is considered. This problem is actual due the need to create parallel programs that effectively utilize the resources of a cluster computing system. Also, the analysis of delays in the communication environment is helpful for system administrators. They can use it for automated troubleshooting and fine-tuning of computing cluster system software. We propose a special approach of analysis of delays in supercomputer communications. There are clustering algorithm is proposed. This algorithm allows smoothing out overshoots and significantly reduce the amount of stored data of delay values by the way of finding the nodes with similar behavior of delays. Algorithm is based on methods of correlations analysis and K-means clustering algorithm. The research was conducted using K60 supercomputer of Keldysh Institute of Applied Mathematics and supercomputers of Moscow State University.

---

### **Simulations in Problems of Ultrasonic Tomographic Testing of Flat Objects on a Supercomputer (abstract)**

*Sergey Y. Romanov*

This paper presents a computer simulation study on a problem of ultrasonic tomographic imaging of welded joints in flat metal objects. The developed algorithms and supercomputer software for reconstructing sound speed images and identifying defects in welded joints were tested on model problems. A specific feature of welded joint inspection is that in most cases the inspected object is accessible only from a single side. This study investigates a tomographic scheme in which a flat object is sounded from a single side by transducer arrays, and reflections from the flat bottom of the object are taken into account. Various schemes of tomographic imaging are investigated and compared for the cases in which the position of the bottom of the object is known, partially known or unknown. The results of this study showed that taking into account reflections from the bottom is of fundamental importance for high-quality image reconstruction. This method allows us to greatly increase the angular range of sounding and to register the waves transmitted through the object in order to achieve high-precision sound speed image reconstruction. The computations were carried out on “Lomonosov” supercomputer at

Lomonosov Moscow State University. The developed software was optimized on a supercomputer for varying number of MPI processes per node. The optimal setup yielded more than a two-fold performance increase.

---

**SoftGrader: automated solution checking system (abstract)**

*Alexander Sysoyev, Mikhail Krivonosov, Denis Prytov and Anton Shtanyuk*

One of the most effective ways to enhance students' learning in programming disciplines is carrying out practical assignments covering all key topics on their own. Ideally, a student deals with an individual set of tasks. While the relevance of such an approach is self-evident, its practical implementation is no easy matter. The main difficulty encountered by teachers is providing a high-quality assessment of the solutions developed by students on a variety of problems. Therefore, it is extremely important to automate the evaluation process. The present article is devoted to the description of SoftGrader – automated checking system, developed in UNN, which allows performing mass testing of student program correctness, both sequential and parallel programs.

---

**Solving the problem of texture images classification using synchronous distributed deep learning on desktop grid (abstract)**

*Ilya Kurochkin and Ilya Kostylev*

The problem of classifying a large set of texture images using deep neural networks is considered. To reduce the learning time of the neural network, it is proposed to use a desktop grid system. The deep neural network architecture is selected and its implementation is described. A method for organizing deep learning based on the data separation approach and synchronous parameter updates during distributed learning is presented. The features of deep learning on a desktop grid system are discussed and the results of a computational experiment are presented.

---

**Students' favorite parallel programming practices (abstract)**

*Igor Konshin*

The paper describes the students' most favorite parallel programming practices when studying a parallel computing course. Students' learning activities are considered as the motivation to study the features of parallel programming. A presentation of parallel computing theory for shared and distributed memory is introduced in a sequence that allows to make students interested in the practical work. There are also derived estimates of the programs' parallel efficiency which can be directly applied to the described learning activities.

Supported by RFBR, project 18-00-01524.

---

**Supercomputer implementation of a high resolution coupled ice- ocean model for forecasting the state of the Arctic Ocean. (abstract)**

*Leonid Kalnitskii, Maxim Kaurkin, Konstantin Ushakov and Rashit Ibrayev*

The paper describes the construction of the coupled ocean-ice Global model INMIO-CICE-CMF2.0 for predicting the state of water and ice in the Arctic Ocean with high spatial resolution (0.1°). The 3D ocean model INMIO is developed at the Institute of Numerical Mathematics and Institute of Oceanology, Russian Academy of Sciences. The sea-ice model CICE(Community Ice Code) is developed Los Alamos National Laboratory. The models are fully coupled at each four time steps using own software named Compact Modeling Framework (CMF ver.2.0).

Outputs are the surface variables sea level and ice conditions (concentration, thickness, velocity, convergence, strength, etc.) and 3-dimensional maps of current, temperature and salinity.

The main aim of the research is developed the algorithm to find the optimal processor configuration of the coupled ocean-ice model for increase performance and optimizing computer resources usage. This is nontrivial problem because this two models are not uniformed and are used for numerical experiments with thousands processors cores on Supercomputers with shared memory using MPI technology. The theoretical and practical aspects of the problem are discussed.

---

### **Supercomputer modeling of the hydrodynamics of shallow water with salt and heat transport (abstract)**

*Alexander Sukhinov, Alexander Chistyakov, Vladimir Litvinov, Asya Atayan, Alla Nikitina and Alena Filina*

The paper covers the investigation of hydrodynamics processes in shallow water taking into account the salt and temperature transport on supercomputer.

Proposed model is based on the equations of motion (Navier-Stokes), the continuity equation for the case of variable density, as well as the equations of transport of salts and heat.

The complex geometry of the coastline and bottom surface of natural reservoirs involves solving a number of problems related to the selection of parameters and the construction of the computational grid used for the numerical implementation of a discrete analogue of the mathematical model of hydrodynamic processes.

Improving the accuracy of calculations, fulfilling the conditions of convergence and stability of the numerical solution of the problem involves crushing the step of the computational grid, which leads to the need to use supercomputer computing systems to obtain calculation results for a given period of time.

The developed numerical methods and algorithms formed the basis of a software package tested on a supercomputer cluster using OpenMP and MPI technologies.

Algorithms are being developed that adapt individual elements of the software package for the NVIDIA CUDA architecture and cloud computing.

Supported by RFBR, project 19-31-51017.

---

### **Supercomputer Simulations in Development of 3D Ultrasonic Tomography Devices (abstract)**

*Alexander Goncharsky and Sergey Seryozhnikov*

This study aims to determine the optimal characteristics of ultrasound tomographic scanners for differential breast cancer diagnosis. Numerical simulations of various tomographic schemes were performed on a supercomputer. The parameters of simulations were matched to those of physical experiments. One of the most important problems in designing a tomographic scanner is the choice of a tomographic examination scheme. The layer-by-layer (“2.5D”) scheme is the most widely used in medical and industrial tomography. In this scheme, the inverse problem is solved for each 2D plane separately. A fully 3D tomographic scheme is an alternative variant, in which the acoustic properties of the object are reconstructed as 3D images. This study shows that the layered 2.5D scheme has limited capabilities and cannot be used for precise tissue

characterization in a general case. The paper presents a comparison of 2.5D and 3D image reconstruction methods in terms of vertical and horizontal resolution, computational complexity of the methods and technical parameters of tomographic scanners. The inverse problem of tomographic image reconstruction is posed as a coefficient inverse problem for the wave equation. The reconstruction algorithms are designed for GPU clusters.

---

**The Algorithms Properties and Structure Study as a Mandatory Element of Modern IT Education (abstract)**

*Alexander Antonov and Vladimir Voevodin*

The paper describes the system of mass practical assignments formed in the framework of teaching the course "Supercomputing Simulation and Technologies" at the Faculty of Computational Mathematics and Cybernetics at the Lomonosov Moscow State University. The practical assignments were held for four years, from 2016 to 2019, and each year about 200 students of the second year of the magistracy passed through them. These practical assignments are aimed at developing skills in the effective use of parallel computing systems. The basics of performing the described assignments is the use of the concept of the information structure of algorithms and programs. This concept is extremely important for writing effective parallel programs, so it should be the cornerstone of modern IT education.

---

**The Numerical Simulation of Radial Age Gradients in Spiral Galaxies (abstract)**

*Igor Kulikov, Igor Chernykh, Dmitry Karavaev, Viktor Protasov, Vladislav Nenashev and Vladimir Prigarin*

In this paper, will present the new results of mathematical modeling of the two-arms galaxy formation and radial age gradients in spirals. The numerical model include self-gravity hydrodynamics equation for gas component of galaxy and collisionless Boltzmann equation for stellar component. To numerical model include important sub-grid physics: star-formation, supernova feedback, stellar wind, cooling and heating function, and non-equilibrium chemistry to ion helium hydride.

Supported by RFBR, project 18-01-00166.

---

**Towards High Performance Relativistic Electronic Structure Modelling: The EXP-T Program Package (abstract)**

*Alexander Oleynichenko, Andrei Zaitsevskii and Ephraim Eliav*

Modern challenges arising in the fields of theoretical and experimental physics require new powerful tools for high-precision electronic structure modelling; one of the most perspective tools is the relativistic Fock space coupled cluster method (FS-RCC). Here we present a new extensible implementation of the FS-RCC method designed for modern parallel computers. The underlying theoretical model, algorithms and data structures are discussed. The performance and scaling features of the implementation are analyzed. The software developed allows to achieve a completely new level of accuracy for prediction of properties of atoms and molecules containing heavy and superheavy nuclei.

---

**Transient Halo in Thin Cloud Layers: Numerical Modeling (abstract)**

*Yaroslav Ilyushin*

In this paper we investigate time-dependent backscattering halo of a pulsed light beam in a layer of scattering medium.

We simulate the polarized radiative transfer in the layer numerically with the three-dimensional upwind-difference scheme. This immediately reveals the dynamic halo effect which we investigate.

Finally, we validate our results against Monte-Carlo radiative transfer simulations and analyze time-dependent structure of the light field using the simulation results.

Supported by RFBR, projects 13-02-12065, 15-02-05476.

---

### **Tuning ANNs Hyperparameters and Neural Architecture search using HPC (abstract)**

*Kamil Khamitov, Nina Popova, Yuri Konkov and Tony Castillo*

Rapid development of deep ANNs made the number of hyperparameters constantly grow. As a sequence various aspects of ANNs, such as inference time, efficient resources utilization, losses and even training time were strongly influenced.

In general methods of hyperparameters tuning are used for adapting well-known ANN models to new tasks or to tasks in similar areas without pre-training or for synthesis of new particular architectures for particular problems. In this article we compare different types of hyperparameters tuning like CoDeepNEAT, Naive Evolution, Tree-Parzen estimation, structured annealing with MorphNet post-tuning. We apply these methods to particular network architectures for image processing and HRM signal estimation. The process of adaptation this technology to big networks requires a lot of computational resources, so it's necessary to use parallel implementations. It can be done by utilizing HPC with hybrid computational nodes. Also we propose new type of tool based on Microsoft NNI. It is used for tuners comparison, convergence analysis, and runs different tuners in parallel mode on cluster nodes.

Supported by RFBR, project 20-07-01053.

---

### **Численное моделирование задач высотной аэродинамики с помощью программного комплекса "Несветай" (аннотация)**

*Владимир Титарев*

В последние 10 лет автором доклада разрабатываются методы численного решения кинетического уравнения с приближенными интегралами столкновений. Данные методы реализованы в пакете программ "Несветай". Отличительные особенности пакета "Несветай" состоят в возможности использования произвольных расчетных сеток в фазовом пространстве, применении экономичной неявной схемы высокого порядка аппроксимации и программная реализация в рамках двухуровневой модели параллельных вычислений MPI + OpenMP. Настоящий доклад посвящен обзору актуальных возможностей кода по численному моделированию задач высотной аэродинамики. Будут приведены примеры расчета внешнего обтекания тел сложной пространственной формы на суперкомпьютерах НИВЦ МГУ им. М.В. Ломоносова, МСЦ РАН и СПбПУ Петра Великого.

При поддержке РФФИ, проекты 18-08-00501 и 18-07-01500.

## Содержание

<b>Полные и короткие статьи .....</b>	<b>3</b>
A Software Package for Studying Dynamical Systems Using the Lyapunov Characteristic Exponent in a Hybrid Computing Environment <i>Sergei Vostokin and Anton Doroshin</i> .....	4
System of intelligent support for solving geophysical problems <i>Pavel Titov, Boris Glinskiy and Gennadiy Zhernyak</i> .....	11
Архитектура, характеристики и функциональные возможности коммуникационной сети Ангара второго поколения <i>А.С. Симонов, А.С. Семенов, Д.В. Макагон</i> .....	19
Вызовы цифровой экономики для науки и образования <i>Л.Н. Щур, С.К. Шикота</i> .....	30
Генерация тетраэдральных сеток для суперкомпьютерного моделирования обтекания аэрокосмических объектов <i>М.К. Ермаков</i> .....	38
Исследование масштабируемости апекс-метода для решения сверхбольших задач линейного программирования на кластерных вычислительных системах <i>Л.Б. Соколинский, И.М. Соколинская</i> .....	49
Многокритериальная оптимизация модели гребного винта с учетом влияния корпуса судна <i>М.П. Лобачев, В.Н. Новоселов, Т.И. Сайфуллин, А.Е. Таранов</i> .....	60
Моделирование влияния системы мониторинга производительности на выполнение коллективных MPI операций <i>А.А. Худолеева, К.С. Стефанов</i> .....	74
Новая магистерская программа «Когнитивные системы» <i>Н.Ю. Золотых</i> .....	84
От «Атомного проекта» к Всемирной Глобальной Научной Программе «Будущее Земли»: математика, компьютеринг, космос <i>Т.А. Сушкевич, С.А. Стрелков, С.В. Максакова</i> , .....	92
Сравнение производительности пакетов симуляции квантовых вычислений QuEST и Intel-QS <i>А.В. Линева, П.Е. Ведруков, Д.С. Куландин, И.Б. Мееров, С. Денисов</i> .....	101
Статически-детерминированный метод прогнозирования динамических характеристик параллельных программ <i>А.А. Клейменов, Н.Н. Попова</i> .....	109
<b>Аннотации стендовых докладов.....</b>	<b>118</b>
Improving performance of linear solvers by using indices compression for storing sparse matrices <i>Sergey Zarechnev and Boris Krasnopolsky</i> .....	119

Высокопроизводительная технология сопоставления изображений на основе оптического потока и метода согласованных оценок <i>К.Г. Пугачев, А.П. Котов, В.А. Фурсов</i> .....	121
Использование MPI- и GPU-ускоренного комбинированного метода квантовой механики, молекулярной механики и метадинамики для моделирования стадии окисления субстрата в каталитическом цикле 2-гидроксибифенил-3-монооксигеназы из <i>Pseudomonas azelaica</i> <i>К.Е. Копылов, Е.М. Кирилин, В.К. Швядас</i> .....	123
Использование вычислений в пониженной точности при реализации метода квантовых траекторий <i>С.В. Боганов, В.Д. Волокитин, И.Б. Мееров</i> .....	126
Новый курс по параллельной обработке графов <i>А.Ю. Пирова</i> .....	129
Опыт применения механизма областей для поэтапного распараллеливания программных комплексов с помощью системы SAPFOR <i>А.С. Колганов</i> .....	131
Особенности трубок чувствительности волн РКІКР и РКІКР во внутреннем ядре Земли <i>О.А. Усольцева, В.М. Овчинников</i> .....	133
Применение алгоритмов машинного обучения для подбора параметров методов решения систем линейных алгебраических уравнений <i>А.А. Петрушов, Б.И. Краснопольский</i> .....	135
Реализация метода конечных разностей во временной области с использованием вычислений в двойной, одинарной и половинной точности <i>А.Н. Арисова, В.Д. Волокитин, И.Б. Мееров</i> .....	137
Симулятор системы управления суперкомпьютерными заданиями как научный сервис <i>А.В. Баранов, Д.С. Ляховец</i> .....	140
Сравнение алгоритмов огрубления графа в задаче параллельного переупорядочения разреженной матрицы <i>А.Ю. Пирова, А.А. Оболенский, В.В. Девликамов, А.Ю. Нестеров</i> .....	142
Сравнение производительности вывода глубоких нейросетевых моделей при разных режимах запуска в Intel Distribution of OpenVINO toolkit <i>А.Е. Хватов, В.Д. Кустикова</i> .....	145
Сравнительный анализ 3D-структур белков на GPU: ре-имплементация алгоритма MATT на CUDA <i>И.А. Тимохин, Д.А. Суплатов, Н.Н. Попова, В.К.Швядас, Вл.В.Воеводин</i> .....	147
<b>Аннотации полных статей на английском языке и приглашенных докладов.....</b>	<b>149</b>
<b>Содержание.....</b>	<b>169</b>

Научное издание

**СУПЕРКОМПЬЮТЕРНЫЕ ДНИ В РОССИИ**

Труды международной конференции

*21–22 сентября 2020 г. Москва*

Издательство «МАКС Пресс»

Главный редактор: *Е. М. Бугачева*

Обложка: *М. А. Еронина*

Напечатано с готового оригинал-макета

Подписано в печать 09.10.2020 г. Формат 60х90 1/8.

Усл.печ.л. 21,5. Тираж 10 экз. Изд. №. 156.

Издательство ООО «МАКС Пресс». Лицензия ИД N 00510 от 01.12.99 г.  
119992, ГСП-2, Москва, Ленинские горы, МГУ им. М.В. Ломоносова,  
2-й учебный корпус, 527 к. Тел. 8(495) 939-3890/91. Тел./Факс 8(495) 939-3891.

Отпечатано в полном соответствии с качеством  
предоставленных материалов в ООО «Фотоэксперт»  
115201, г. Москва, ул. Котляковская, д.3, стр. 13.

