Supercomputer technologies for solving problems of computational physics

Boris Glinskiy¹, Yury Zagorulko², Igor Kulikov¹ and Anna Sapetina¹

¹ Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Akad. Lavrentieva pr. 6, Russia
² A.P. Ershov Institute of Informatics System SB RAS, Novosibirsk, Akad. Lavrentieva pr. 6, Russia
E-mail: gbm@sscc.ru

Abstract. The issues of using supercomputer technologies developed by the authors to solve compute-intensive problems of mathematical physics are discussed. The technology of developing algorithms and software for supercomputers contains the three related stages: co-design, by which we understand the adaptation of the problem statement, the mathematical method and the computational algorithm to the parallel architecture of the supercomputer at all stages of the problem solution; studying the scalability of computational algorithms for the most promising supercomputers based on simulation modeling; evaluation of the energy efficiency of algorithms for various implementations on a given supercomputer architecture. It is proposed to further develop this approach with the use of the intelligence support for solving compute-intensive problems using the ontology of computational methods and algorithms for solving the problems of computational physics, the ontology of computational heterogeneous architectures and decision rules.

1. Introduction
The state-of-the-art in the of supercomputer development is characterized by the emergence of a variety of projects on the creation of an exascale-class supercomputer. Thus far, developments in the field of exascale supercomputers are conducted by different teams of developers in the United States. The collaboration in this direction is carried out, for example, by the national laboratories of the US Department of Energy: Sandia and Oak Ridge. In Europe, there are also similar programs; seven European countries have signed the declaration of the Joint Project EuroHPC aimed at the creation of exascale supercomputers. In Japan (the RIKEN institution), the development of a supercomputer has already begun, where it will be assembled and installed. There are numerous international projects to develop the system and the application software for exascale-class supercomputers with the participation of the United States, the countries of the European Union, Japan, China, Russia (IESP, G8 EXASCALE, CRESTA, etc.). In [1-3], a review and various approaches to the exascale scientific software design are given. In [4], the problems typical of exascale systems are listed, such as insufficient concurrent work available to maintain an extensive utilization of all resources, a time-distance delay intrinsic of parallel actions and resources on the critical execution path, which is not necessary in a sequential version, the delay due to the lack of availability of shared resources. In addition, in [4] various methods for overcoming the above-mentioned problems are discussed.
The current state of the computer technologies provides the researchers with a wide range of computational architectures with the ability to use high-performance parallel computing to solve their problems. It should be noted that along with classical computational clusters with MPP (massive parallel processing), SMP (symmetric multiprocessing) architectures, hybrid supercomputers equipped with accelerators of computing, are becoming increasingly popular. In the TOP 500 of the world’s most powerful computers (www.top500.org) a supercomputer Summit with a capacity of 188 PFlops is in the first place. It includes 4068 computer nodes, each containing two 22-core processor IBM Power9 and six GPUs NVIDIA Tesla V100, linked by a dual-channel Mellanox EDR. Summit has also more than 10 PB of RAM combined with fast high-bandwidth channels for the efficient data transfer. Note that this supercomputer is optimized for the analysis of big data; nevertheless, intensive work on using the advantages of the V100 architecture for solving problems of plasma physics and accelerator physics is underway.

The performance of the most powerful Russian supercomputer Lomonosov-2 is 2.5 PFlops. However, in Russia, it is planned to build up the national supercomputer centers with a capacity of dozens PFlops. On such supercomputers, it is possible to solve problems of essential computational complexity.

In each subject area, where there is a set of methods for solving compute-intensive problems on modern multi-core computing systems, it is important to choose a method that would most effectively solve the problem in a reasonable time with a given accuracy. On the other hand, the representation of knowledge in the form of ontologies and inference rules for solving specific problems is becoming increasingly popular [5,6]. Note that ontologies (a system of concepts and relationships linking them) are the foundation of knowledge bases in complex areas that require systematization. The application of the ontologies facilitates the creation of generally available knowledge bases and the use of artificial intelligence methods thanks to a large set of freely available editors for creating ontologies and inference engines (reasoners) that make logical conclusions.

Thus, at present, the following situation is emerging: a researcher on the one hand, has a set of computational methods and algorithms that solve a complex computational problem, and, on the other hand – a set of computational architectures, libraries and software tools to solve the problem. Considering that it is difficult for a researcher to competently navigate both in computational methods for solving a problem and in existing means for high-performance computing (architectures, parallel languages and software tools), it is necessary to create a system of intelligence support (IS) for making decisions with allowance for a the features of the simulated environment to use a computing architecture.

It should be pointed out that numerical algorithms are being developed slower than hardware. Therefore the existing algorithms and programs for solving compute-intensive problems will be applied at the first stage of using petascale- and exascale supercomputers, containing hundreds of millions of cores.

Consequently, the creation of modern computing technologies for the parallel use of such a large number of computing cores is an urgent task.

2. A supercomputer technology for solving compute-intensive problems based on an integral approach

We have offered the integrated approach to the development of algorithms and software for petascale- and exascale-class supercomputers [7]. It contains the three stages.

The first stage is the co-design, which is based on the development of a parallel computational technology, with allowance for all aspects of parallelism. The co-design of parallel methods for solving large-scale problems is difficult to formalize. It is impossible to make a collection of recipes for the efficient solution of any problem. However, some general approaches can be proposed. The co-design approach concept consists of the following steps, with allowance for
the target hardware/software platform: the formulation of physical statement of the problem, the mathematical formulation of the physical problem, the development of the numerical methods, the selection of data structures and parallel algorithms, the consideration of a supercomputer architecture, the usage of code optimization tools.

We use an extended definition of the co-design in contrast to the common conception which consists in the joint development of software and hardware. In such an approach, not only the comparison of the problem solution methods is becoming relevant but also the comparison of the efficiency of using various physical and mathematical statements [8, 9].

The second stage is the anticipated development of algorithms and software for the most promising exascale supercomputers. This stage is based on the simulation of the algorithm behavior within a certain supercomputer architecture. For the simulation of distributed systems it is best to use distributed simulation based on message passing.

The multi-agent approach [10] is used for the simulation of parallel programs run on a large number of cores due to such properties as decentralization, self-organization, and intelligent behavior. Among many multi-agent simulation platforms, the adaptable distributed simulation system AGNES [11] was chosen. It was successfully used in the scalability study of a series of parallel problems when executed on a large number of cores [12].

The third stage is estimating the energy efficiency of the algorithm with different implementations for a single architecture or for different supercomputer architectures. In this paper, the term "energy efficiency" for scientific HPC applications means the most efficient use of each core, a processor or a computational accelerator; the minimization of communications among computational nodes; a good workload balancing of the program. The minimization of communications enables a decrease in the idle standing time for processors and accelerators. The good workload balancing enables a uniformly load of a computational system. The most energy efficient algorithm gives the best FLOPS per Watts (Joules/sec) value. The efficiency of this approach is illustrated on some examples of complex computing problems in seismology, plasma physics and astrophysics.

Let us consider the problem of numerical simulation of the propagation of elastic waves in heterogeneous three-dimensional media with complex subsurface geometries, which is a difficult problem from the point of view of computation, which requires the use of effective methods of parallelizing and scaling algorithms.

Quite often, the topography of various real geophysical objects does not allow one to maintain an observational system. Therefore, constructing their 3D models requires solving the inverse problem by solving a set of direct problems: for different values of the elastic parameters of a heterogeneous medium; for various geometries of objects composing a model.

We apply the above-discussed approach to solving the problem of seismic wave propagation in a heterogeneous medium typical of magmatic volcanoes. Both active and sleeping volcanoes are potentially dangerous to the environment due to the possibility of sudden disaster eruptions. Using the methods of the active vibroseismic monitoring of magmatic structures will allow predicting a probable time of an eruption [13].

For the purpose of the co-design, we have made a comparison of the developed parallel implementations of solutions to the elastodynamic problem written in terms of the velocities of displacement and stress and in terms of displacements for the computational clusters, equipped with graphics cards. The simulation domain is considered to be an isotropic 3D-inhomogeneous elastic structurally complex medium which is a parallelepiped, one of whose sides is a free surface.

At the step of designing a numerical method, the most "flexible" and widespread technique for solving a three-dimensional elastodynamic problem is a finite difference method. Let us preliminarily notice that explicit finite difference schemes fit the architecture of graphics accelerator, because they are directly mapped to the topology of GPU architecture, and involve independent computations of values at each step and in each cell of the computational domain.
In order to numerically solve elastodynamics equations in terms of the displacement velocities and the stress we apply the well-known Verrier finite difference scheme on a staggered grid. The calculation of its difference coefficients is based on integral conservation laws. To solve the problem in terms of displacements, we use a similar finite difference scheme [13]. We should note the main difference between the algorithms, which can be constructed on the above-mentioned finite difference schemes. The calculation of the displacement velocities and the stress requires a larger memory size (at least, 18 3D arrays with the unknowns should be stored), but requires a smaller number of float-in-point operations in total (57 operations for calculating the values in a cell for one time step). The calculation of the displacements requires a smaller memory size (at least, six 3D arrays with unknowns), but a larger number of operations (98 operations for one time step).

The numerical experiments have shown that the time of the calculations of the displacements and the time of the calculations of the displacement velocities and the stress at an equal number of nodes is roughly the same in spite of the fact that the displacements calculation is performed with a larger amount of floating point operations at each time step. Therewith, the displacement calculations requires almost half as much of the GPU memory size. Based on the results obtained we prefer using the approach proposed to calculating the displacements. Figure ?? shows a diagram of the vibroseis monitoring of the Elbrus volcano, its geophysical model and wave field propagation. The results of numerical experiments on the NCS-30T cluster have shown that it is possible to distinguish kinematic differences in the wave field and, approximately, restore the magma level in the output channel.

![Volcano Elbrus Monitoring scheme Wave field](image)

**Figure 1.** The scheme of vibroseismic monitoring of the Elbrus volcano and the result of the numerical experiment.

The scalability of the resulting algorithms has been tested using the AGNES simulation system. We have considered utilizing the hybrid supercomputers equipped with GPU or Xeon Phi architectures for solving the elastodynamic problem. The results of the research into the numerical simulation scalability of the elastic wave propagation problem in complex media are presented in Figure ???. The two algorithms that solve this problem in two ways have been studied. From the figures it is clear that the scalability of the two approaches slightly differs, and both algorithms are suitable for execution on a large number of cores. The real calculations for the verification were performed with the Siberian Supercomputer Center.
The further research direction we see in the creation of methods and systems of intelligence support (IS) for solving compute-intensive problems on supercomputers.

3. The basic principles of building the IS system
Consider the principles of building the IS system for solving compute-intensive problems and its main blocks. Figure 3 shows a conceptual scheme of the IS system.

The core of such a system is a knowledge base, including a set of interconnected ontologies and inference rules that extend the logic of these ontologies. An important module of the IS system is an inference engine (a reasoner), which, using the knowledge base and the user specification of a problem, builds an optimal scheme for its solution.

Another component of the IS system, i.e. the automated code generation module, supports the creation of a parallel code that solves the problem set by the user. This module inserts the corresponding code fragments from the software component library (SC library) into a scheme of solving the problem. If there is no suitable component in the SC library, the user can substitute a necessary component by taking it from the standard library or writing a new one.
The IS system also includes a simulation modeling unit that evaluates the scalability of the resulting code on a given architecture before carrying out the final time-consuming calculations on a multi-core computing system. The main purpose of this unit is to select the optimal number of cores for the implementation of the algorithm by studying its behavior in the model time.

4. Conclusion
This paper considers the use of the integrated approach to the development of algorithms and software for the numerical simulations of seismic wave propagation in the media with complex subsurface geometries on a hybrid cluster. For the purpose of the co-design, we have made a comparison of the developed parallel implementations of solutions with the elastodynamic problem written in terms of the displacement velocities and the stress and in terms of displacements for the computational clusters, equipped with graphics cards. We have studied the execution time of the created parallel programs and their scalability (including the simulation of program execution on a large number of cores by using the simulation system AGNES).

In this paper, we have also presented the concept of the IS system for solving compute-intensive problems of mathematical physics. The paper shows how, having, on the one hand, the ontology of computational methods and the corresponding parallel algorithms, and on the other hand, the ontology of parallel architectures and technologies, it is possible to create parallel programs for solving problems as soon as possible.

Acknowledgments
This research was conducted within the framework of the budget project No. 0315-2019-0009 for ICMMG SB RAS and supported in part by the Russian Foundation for Basic Research [grant No. 19-07-00085].

References
[1] Reed D A and Dongarra J. 2015 Commun. ACM 58(7) 56–68
[2] Keyes D E 2011 C.R. Mechanique 339 70–77
[3] Asanovic K et al 2009 Commun. ACM 52 56–67
[4] Sterling T 2013 Adv. Parall. Comp 24 104–117
[5] Compton M et al 2012 Web Semant. Sci. Serv. Agents World Wide Web 17 25–32
[6] Keet C M, Lawrynowicz A, dAmato C, Kalousis A, Nguyen P, Palma R, Stevens R and Hilario M H 2015 Web Semant. Sci. Serv. Agents World Wide Web 32 43–53
[7] Glinskiy B, Kulikov I, Chernykh I, Snytnikov A, Sapetina A and Weins D 2017 CCIS 793 278–289
[8] Glinskiy B, Kulikov I, Snytnikov A, Romanenko A, Chernykh I and Vshivkov V 2014 Supercomputing Frontiers and Innovations 1(3) 88–98
[9] Glinsky B, Kulikov I, Chernykh I et al 2017 LNCS 10049 342–353
[10] Bellifemine F L, Caire G and Greenwood D 2007 Developing Multi-Agent Systems with JADE (Chichester: Wiley)
[11] Podkorytov D, Rodionov A and Choo H 2012 Proc. of the 6th Int. Conf. on Ubiquitous Information Management and Communication (Kuala Lumpur: ACM) p 115
[12] Glinsky B M, Marchenko M A, Mikhailenko B G, Rodionov A S, Chernykh I G, Karavaev D A, Podkorytov D I and Vins D V 2013 Simulation Modeling of Parallel Algorithms for Exaflop Supercomputers (in Russian) Inf. Technol. Comput. Syst. 4 3–14
[13] Boris Glinskiy, Anna Sapetina, Valeriy Martynov, Dmitry Weins, Igor Chernykh 2017 CCIS 753 261–274