Ontological Approach to Providing Intelligent Support for Solving Compute-Intensive Problems on Supercomputers

Galina Zagorulko[1(✉)], Yury Zagorulko[1], Boris Glinskiy[2], and Anna Sapetina[2]

[1] A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, Acad. Lavrentjev avenue 6, 630090 Novosibirsk, Russia
gal@iis.nsk.su

[2] Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of the Russian Academy of Sciences, Acad. Lavrentjev avenue 6, 630090 Novosibirsk, Russia

Abstract. The paper describes an approach to providing intelligent support for solving compute-intensive problems on supercomputers, based on the ontologies and decision rules helping the users choose a computational method suited best to their task and supercomputer architecture. The authors focus on the concept and following components of intelligent support: the ontology of the problem area "Solving Compute-intensive Problems of Mathematical Physics on Supercomputers", an information-analytical Internet resource based on this ontology, providing the users access to information necessary for solving compute-intensive problems on supercomputers and an expert system helping users develop a parallel code based on ready-made software components. The paper describes in detail the conceptual scheme of intelligent support and the ontologies developed. To avoid possible errors in designing an ontology, in model means for knowledge representation absent in the ontology description language, to systematize and facilitate populating an ontology with concept instances, we have developed and applied a number of structural and content patterns.

Keywords: Supercomputers · Intelligent support · Ontology · Mathematical physics · Decision support system

1 Introduction

Currently, supercomputer technologies are widely used in the mathematical modeling of various physical phenomena and processes. However, the development of programs for supercomputers is becoming more complicated with an increase of parallelism and use of heterogeneous architectures. In this connection, creation of intelligent support means for solving compute-intensive problems on both modern and future supercomputers is an emerging priority.

The main idea of the intelligent support (IS) for solving computationally complex tasks of mathematical physics on supercomputers is using the knowledge of a problem

S. O. Kuznetsov and A. I. Panov (Eds.): RCAI 2019, CCIS 1093, pp. 363–375, 2019.
https://doi.org/10.1007/978-3-030-30763-9_30

area (PA). This includes the comprehensive knowledge of the methods and algorithms for solving such problems on supercomputers and their implementations, as well as the experience of solving such problems presented in the form of techniques and software components (fragments of parallel code).

As a conceptual basis of intelligent support, we propose to use a system of interrelated ontologies, including the ontology of computational methods and parallel algorithms, as well as the ontology of parallel architectures and technologies.

To ensure the success of solving tasks on the supercomputer, the users need information support: convenient access to the information on the available methods and algorithms, on the capabilities and limitations of each of them, and on the characteristics of their implementations. The means of such support is the information-analytical resource created on the basis of the above ontologies, supplied with an advanced user web-interface providing content-based access to this kind of information.

At the stage of developing a code to solve the user's problem, component support plays an important role. The opportunity to choose ready-made software components implementing the necessary algorithms can significantly simplify and speed up developing the code executed on a supercomputer. A means of supporting the solution of this problem is a library of previously developed software components equipped with unified specifications, which can serve as a basis for integrating the components into a parallel code.

Another means of intelligent support is an expert system, which should help the user to choose the implementation of a computational method and a supercomputer architecture optimal for his task.

The paper describes the ontological approach to providing intelligent support for solving compute-intensive problems on supercomputers. Section 2 provides an overview of current information resources that support solving tasks on supercomputers. Section 3 details the concept of intelligent support. Section 4 describes the ontologies underlying the IS.

2 Overview of Information Resources Supporting the Solution of Tasks on Supercomputers

At present, there are several projects in Russia developing and supporting solving tasks on supercomputers. However, their support is limited to systematizing the knowledge on this topic and providing access to it.

The most significant Russian project is AlgoWiki [1], which provides an information resource positioned on the Internet as an open encyclopedia on algorithms' properties and peculiarities of their implementation on various software and hardware platforms, from mobile to exaflops supercomputer systems enabling collaboration with the entire world community.

The stated objective of the AlgoWiki resource is to give a comprehensive description of each algorithm, which will help to assess its potential in relation to a particular parallel computing platform. For this, AlgoWiki provides each algorithm with a detailed description, including a description of its parameters necessary for serial

and parallel numerical implementation, indicating the most time-consuming parts. In addition, AlgoWiki provides references to ready-made packaged solutions.

The second most important Russian web resource is parallel.ru [2]. This resource is aimed at informing the users about all important events and updates in the supercomputer community (new technologies, software products and tools for developing parallel code, conferences, new supercomputers, latest news).

The above-mentioned resources systematize information about their subject areas without resorting to ontologies, which reduces their capabilities dramatically, both in terms of knowledge and data presentation and ease of access to them. Nevertheless, the presentation of information about the subject area, covering the problems of solving compute-intensive problems on supercomputers, in the form of ontologies can provide not only convenient access to it, but also support the user in choosing the optimal algorithms and parallel architectures in solving his applied problems due to the possibilities of logical inference on the ontology.

The need to create resources based on ontological descriptions, for various fields of science (physics [3], geology [4], biology [5], astrophysics [6, 7], etc.) has been also discussed abroad. Currently, there are three web resources on astrophysics [8], on genomics [9], on geology and geophysics [10]. However, similarly to the Russian resources, they do not provide to the user the comprehensive support, that will be provided by the resource proposed in this work.

The authors also explore how ontologies can be used to improve the efficiency of computational resources and support the users who solve their problems on them.

For example, in [11], it is suggested that ontologies can be used to describe Grid resources, which would simplify and structure the construction of Grid applications by composing and reusing existing software components and developing knowledge-based services and tools. For this purpose, an ontology of the Data Mining knowledge area has been developed. This ontology makes it easier to develop Grid applications focused on identifying knowledge from data. It offers experts in this area a basic (reference) model for various data analysis tasks as well as methodology and software developed to solve these tasks, and helps the user in choosing the optimal solution.

In [12], authors propose an approach that uses an ontology describing the metadata of a Grid resource and the reference mechanisms on it for more efficient management of Grid resources.

[13] discusses the use of ontologies and ontology reference mechanisms to assist users in solving compute-intensive problems on heterogeneous computing architectures, in particular on clusters equipped with NVIDIA GPGPUs and Xeon-Phi coprocessors in addition to the traditional Intel processor. Here, ontology inference mechanisms help to find the best possible solution by combining hardware, software, and planning strategies. The paper demonstrates the application of the approach to solving bioinformatics problems.

The described approaches to the use of ontologies are close to our approach, but they focus on other areas of knowledge and architectures.

3 The Concept of Intelligent Support

As mentioned above, intelligent support for solving compute-intensive problems of mathematical physics on supercomputers is based on using knowledge of this problem area presented in the form of ontologies and experience in solving such problems, embodied in the form of techniques and parallel code fragments.

Figure 1 shows a conceptual diagram of intelligent support for solving compute-intensive problems.

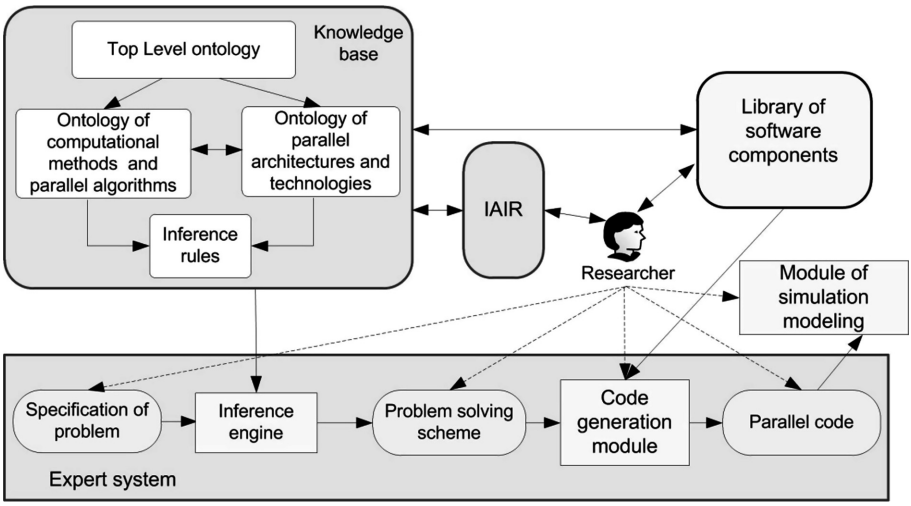

Fig. 1. Conceptual scheme of intelligent support for solving compute-intensive problems.

Intelligent support is provided by the following components: a knowledge base, an information-analytical Internet resource for supporting the solution of compute-intensive problems on supercomputers (IAIR SSCIP), an expert system (ES), a library of software components and a simulation module.

The core of IS is the knowledge base including interrelated ontologies: the top level ontology of the PA "Solving Compute-Intensive Problems of Mathematical Physics on Supercomputers", the ontology of computational methods and parallel algorithms, the ontology of parallel architectures and technologies, as well as inference rules that expand the logic of these ontologies.

The library of software components includes the code fragments implementing the necessary algorithms executed on a supercomputer. Software components are provided with unified specifications, which enable their integration into a parallel code. Also, this library provides the user with ready-made software packages that he/she can use when solving the task.

The knowledge base is used by an information-analytical Internet resource for supporting the solution of compute-intensive problems on supercomputers (IAIR SSCIP) and an expert system (ES) that helps the user in building a parallel code.

The ontology-based IAIR SSCIP is designed to provide the user with information support in solving compute-intensive problems. It provides complete information about the methods and algorithms available, capabilities and limitations of each of them, and characteristics of their implementations. This resource is equipped with an advanced user web-interface providing meaningful access to this kind of information. The user will be able to obtain detailed information necessary to solve his/her task: methods and algorithms currently available, their description and numerical implementation features, tools available for creating a parallel code, available supercomputer architectures and their features, as well as features of programming for these architectures. These capabilities of the IAIR SSCIP minimize the time required for an in-depth familiarization with the problem area, since all the necessary information is structured and collected in one place.

The expert system is designed to assist the user in building a parallel code that solves his/her problem on a supercomputer. The users submit to the ES the specification of their compute-intensive problems, and at the output they get a scheme for solving the problem, and then a parallel code.

In addition to the knowledge base, the ES includes an inference machine (solver) and an automated code generation module.

The inference machine uses the knowledge base and specification of a compute-intensive problem written by the user to build the optimal scheme for solving the problem.

The module of automated code generation supports the creation of the parallel code that solves the problem. This module inserts the corresponding code fragments from the software component library (SC) into the problem solving scheme. If there is no suitable component in the SC library, the user can substitute the necessary component by taking it from a standard library or writing a new one.

The module of simulation modeling [14], which evaluates the scalability of the resulting code, is also included in the intelligent support contour. The main purpose of this module is to select the optimal number of cores for implementing the code by studying its behavior with different numbers of cores in model time.

4 Ontologies

The backbone of the IS knowledge base is the ontology of the problem area “Solving Compute-intensive Problems of Mathematical Physics on Supercomputers”. This multi-level ontology contains several interrelated ontologies; important among them are the ontology of computational methods and parallel algorithms and ontology of parallel architectures and technologies.

Consider the upper level of the ontology of the PA “Solving Compute-intensive Problems of Mathematical Physics on Supercomputers” (see Fig. 2).

The main Objects of research in this field are Physical Objects and Physical Phenomena, which are studied in certain divisions of science and are based on the Fundamental laws of nature derived, in turn, from Experiments and observations. The objects of research are considered in the form of an approximate Physical Model, described by the Mathematical Model, which formalizes the Fundamental laws of

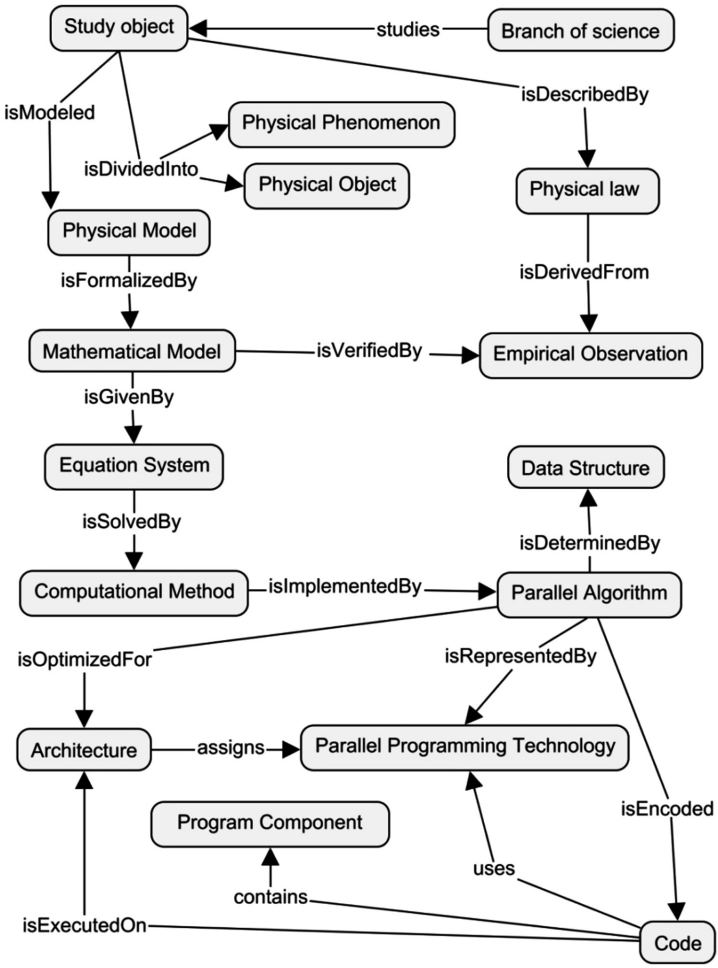

Fig. 2. The top level of ontology of the PA "Solving Compute-intensive Problems of Mathematical Physics on Supercomputers".

nature, is given by a system of equations and verified in the course of Experiments and observations. Consequently, we have a Mathematical model in the form of a System of equations satisfying the initial and boundary conditions containing all parameters (equation of state, values of coefficients, etc.). Since we are considering a limited field of mathematical physics, the system of equations will correspond to this area.

The system of equations is resolved by the Computational Method, which, in turn, is implemented by a Parallel algorithm, determined, among other things, by the Data Structure. The parallel algorithm implementing the Computational Method is optimized for the parallel architecture available to the user and is represented using certain Parallel Programming Technologies. The final representation of a parallel Algorithm is encoded

by a program code consisting of a set of Software components and executed on a parallel Architecture.

The ontology described contains the basic concepts of the PA and their interrelationships. Since concepts such as the Computational Method, Parallel Algorithm, Parallel Architecture are very important for this PA and define the specifics of solving computationally complex problems, their descriptions are discussed in more detail and divided into two separate ontologies: the ontology of computational methods and parallel algorithms (Fig. 3) and ontology of parallel architectures and technologies (Fig. 6).

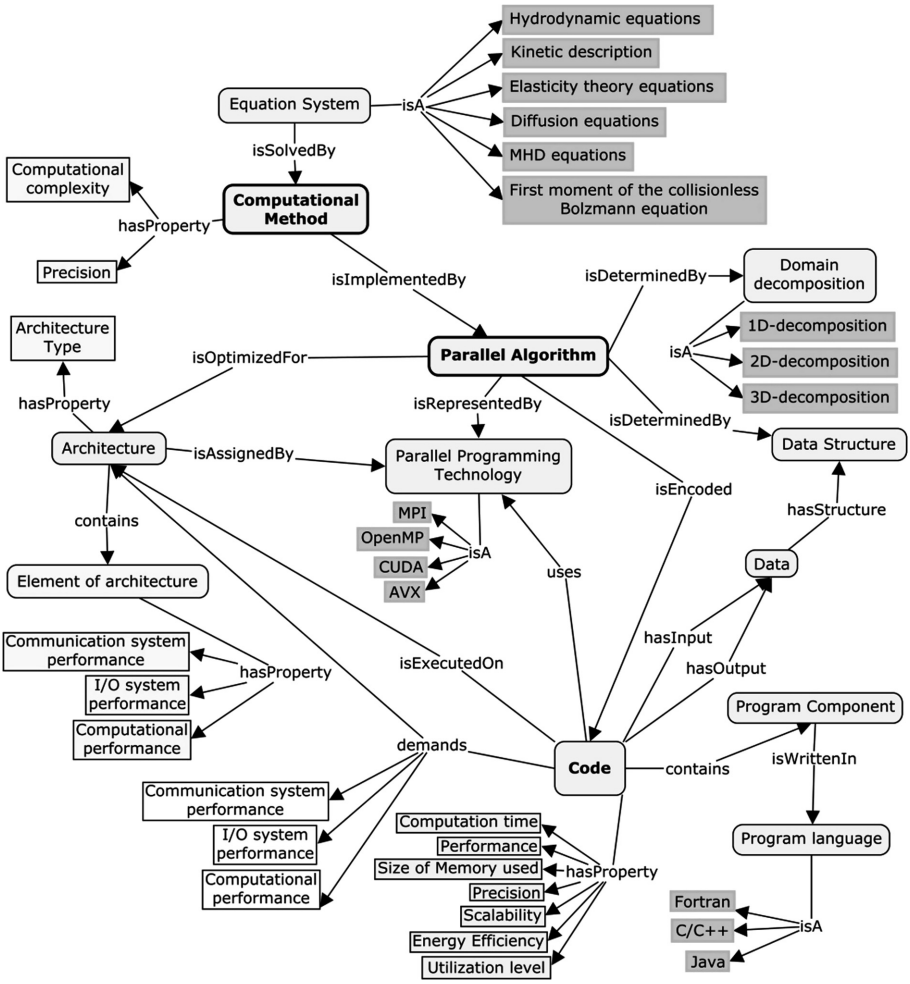

Fig. 3. Ontology of computational methods and parallel algorithms.

To systematize and facilitate populating these ontologies with the instances of their concepts, appropriate ontological design patterns have been developed [15, 16]. Figure 4 presents a structural-content pattern describing the class Code, which is one of the main classes of the ontology of computational methods and parallel algorithms. Class Code is characterized by such interrelated properties as Performance, Computation time, Size of memory used, Precision, Scalability, Energy Efficiency, Utilization level (of memory or processor). The structural-content pattern describing the class Parallel algorithm is shown in Fig. 5.

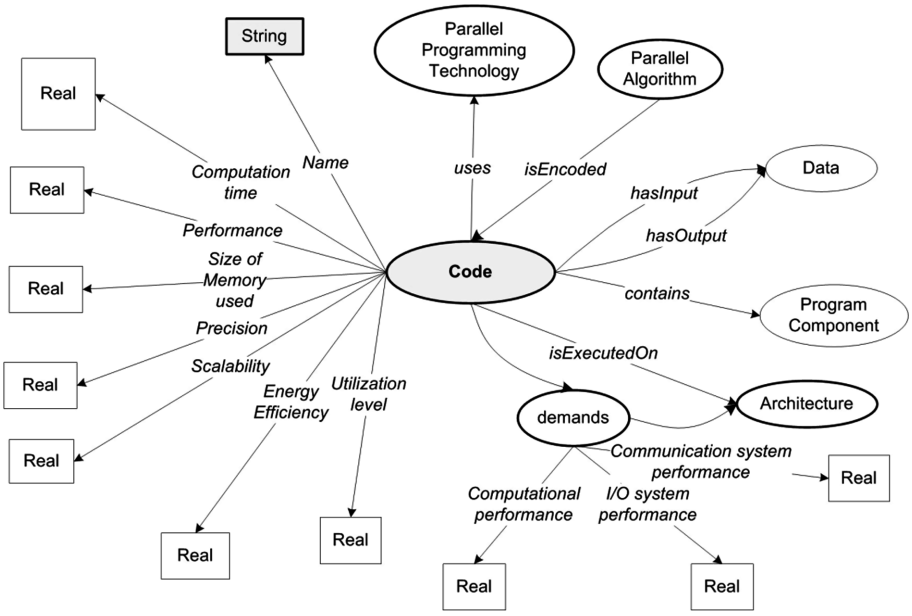

Fig. 4. The structural-content pattern for defining instances of the "Code" concept.

A fragment of the ontology of parallel architectures and technologies is shown in Fig. 6. The main elements of a parallel architecture are Computing Nodes connected by Interconnect. Computing nodes, in turn, contain Computing Devices—the main CPUs and Computing Accelerators, and Memory, coupled with Computing Devices. There are two types of Computing Accelerators. These are General-purpose Accelerators and Accelerators that are specialized for specific tasks.

A specific configuration of the Architecture is determined by the Type of the computing system (the main representatives are MPP, SMP, NUMA). It is characterized by such properties of its Elements as Computational Performance, Communication System Performance, and I/O System Performance (Fig. 7). The Code, executed on a parallel Architecture, imposes certain requirements on the listed properties of the Architecture Elements.

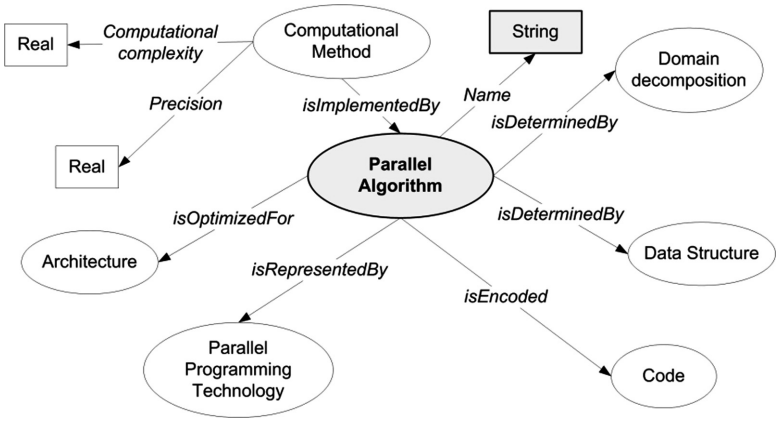

Fig. 5. The structural-content pattern for defining instances of the "Parallel algorithm" concept.

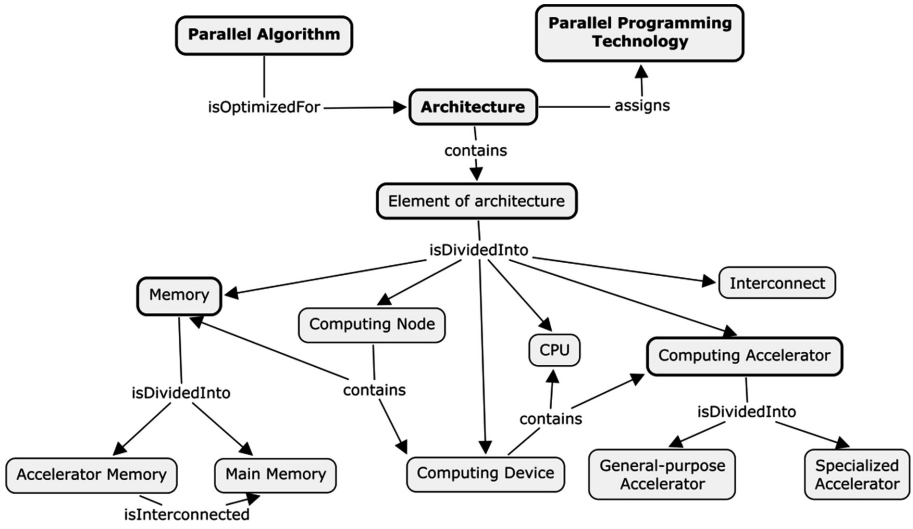

Fig. 6. Ontology of parallel architectures and technologies.

The above ontologies are described in the OWL language [17] and include descriptions of classes and their properties, as well as instances of classes that make up the knowledge base filled with specific methods, algorithms, software components and elements of parallel architectures.

As noted above, to systematize and simplify populating the ontologies with the instances of their concepts, a number of ontology design patterns have been developed (Figs. 4, 5 and 7). In addition, the patterns are used to model the techniques for representing some useful constructions not found in the OWL language. For example, the domain values of the type of a parallel Architecture are represented by the

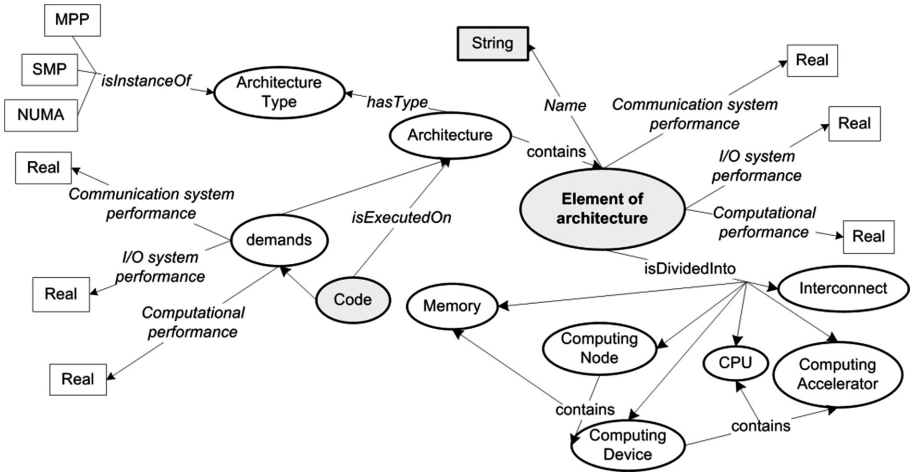

Fig. 7. Structural-content pattern for defining instances of the "Element of architecture" concept.

Architecture Type class with MPP, SMP, NUMA instances. The attributed relation "demands" (i.e. relation with attributes), which connects the "Code" and "Architecture" classes, is specified by the "demands" service class of the same name, its connections with these classes, as well as with the values of the demands (Fig. 7).

To build ontologies and set axioms, the ontology editor Protégé 5.2 with the inference machines was used [18]. The logical inference in OWL-ontologies is performed on the basis of the axioms specified in the ontology, by one of the inference machines (Pellet, FaCT++, HermiT).

In addition to the ontology, the IS knowledge base contains expert rules allowing using the specification of a compute-intensive problem to select software components for generating a code and determining the architecture on which it will be executed, and vice versa. These rules allow you to display the information not explicitly contained in the ontology. For example, on the basis of well-written inference rules and using the Properties of the Computing Method and the Parallel Algorithm, you can define its bottleneck in terms of compute-bound (the limiting factor for solving such a problem is the speed of the processor) and memory-bound problem (the limiting factor is the speed of access to memory).

Further, guided by the correct inference rules, one can give qualitative and quantitative estimates of the Properties of the Code executed on the Architecture, focusing on its Properties. Also, inference rules may allow setting the requirements that the Program Code imposes on the Properties of the Architecture (Fig. 4). For example, if the performance of the communication system of the Architecture is not high enough, efforts to improve the performance of the Code will be in vain. Thus, the inference rules define implicit links between the Computational Method, Parallel Algorithm, Target Architecture, and Code. Guided by them at different stages of problem solving, you can quickly find the optimal approach to the solution of your problem.

To represent the expert rules, the SWRL (Semantic Web Rule Language) [19] is used. (Note that the rules written in the SWRL are, in fact, Horn clauses, which greatly simplifies their construction and understanding.)

Let us give examples of the SWRL rules involved in the choice of the architecture on which the code can be executed.

The first rule checks whether the Computing Device has enough power to execute the code.

Code(?c), Architecture (?a), ComputingDevice (?cd), contains(?a, ?cd),
demands(?d), makesDemandsCode (?d, ?c),
makesDemandsArchitecture(?d, ?a),
demandsComputationalPerformance(?c, ?cpc),
ComputationalPerformance (?cd, ?cpa),
swrlb: greaterThanOrEqual (?cpa, ?cpc)
→ *comparableToComputationalPerformance(?c, ?cd).*

The second rule checks whether all the code demands for the architecture are satisfied and decides whether the code can be executed on the given architecture.

Code (?c), Architecture(?a), ComputingDevice(?cd), contains (?a, ?cd),
IOSystem (?ios) contains (?a, ?ios),
Interconnect (?i), contains (?a, ?i),
comparableToComputationalPerformance(?c, ?cd),
comparableToPerformanceInputOutputSystem (?c, ?ios),
comparableToPerformanceCommunicationSystem(?c, ?i),
→ *isExecutedOn(?c, ?a).*

5 Conclusion

The paper presents an ontological approach to providing intelligent support for solving compute-intensive problems of mathematical physics on supercomputers. Primarily, this support is necessary for regular users, who, as a rule, have little understanding of multiprocessor system architectures and software capability for creating parallel programs. Another group of users who will benefit from the IS are specialists in computational technologies, well versed in supercomputer architectures and parallel languages, but less familiar with computational methods.

Intelligent support is provided by the following components: the ontology of the PA "Solving Compute-intensive Problems of Mathematical Physics on Supercomputers", an information-analytical Internet resource to support solving compute-intensive problems on supercomputers, and an expert system helping users develop a parallel code based on ready-made software components.

It is vital to note that in order to help supercomputer users to create effective programs, it is most important to have detailed ontologies, a fairly complete library of software components implementing parallel algorithms and fragments of parallel code, and a well-chosen set of decision rules formalized as inference rules, which will generate the program code taking into account the specific features of the problems being solved and the available multiprocessor system architectures.

At the moment, the efforts of developers focus on the implementation and improvement of these intelligent support components.

Acknowledgment. This work is financially supported by the Russian Foundation for Basic Research (Grants no. 19-07-00085 and no. 19-07-00762).

References

1. AlgoWiki: Open Encyclopedia of Algorithm Properties. https://algowiki-project.org/ru/. Accessed 05 May 2019
2. parallel.ru. https://parallel.ru. Accessed 05 May 2019
3. Cvjetkovic, V.: Web physics ontology: online interactive symbolic computation in physics. In: Proceedings of 2017 4th Experiment@International Conference (exp.at 2017), Faro, Portugal, pp. 52–57. IEEE (2017)
4. Xiaogang, M.: Ontology spectrum for geological data interoperability. ITC, Netherlands (2011)
5. Cook, D.L., Neal, M.L., Bookstein, F.L., Gennari, J.H.: Ontology of physics for biology: representing physical dependencies as a basis for biological processes. J. Biomed. Semant. **4** (1), 41 (2013)
6. 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
7. Louge, T., Karray, M.H., Archimède, B., Knödlseder, J.: Semantic interoperability in astrophysics for workflows extraction from heterogeneous services. In: van Sinderen, M., Chapurlat, V. (eds.) IWEI 2015. LNBIP, vol. 213, pp. 3–15. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47157-9_1
8. ESPAS. https://www.espas-fp7.eu/portal/browse.html#ontology. Accessed 05 May 2019
9. SemGen. http://sbp.bhi.washington.edu/projects/semgen. Accessed 05 May 2019
10. Awesome geoscience semantics. http://www.geoscience-semantics.org. Accessed 05 May 2019
11. Cannataro, M., Comito, C.: A data mining ontology for grid programming. In: Proceedings of 1st International Workshop on Semantics in Peer-To-Peer and Grid Computing (In Conjunction with WWW 2003), Budapest, Hungry, pp. 113–134 (2003)
12. Amarnath, B.R., Somasundaram, T.S., Ellappan, V.M., Buyya, R.: Ontology-based grid resource management. Softw. Pract. Exp. **39**(17), 1419–1438 (2009)
13. Faheem, H.M., König-Ries, B., Aslam, M.A., Aljohani, N.R., Katib, I.: Ontology design for solving computationally-intensive problems on heterogeneous architectures. Sustainability **10**(2), 441 (2018)
14. Podkorytov, D., Rodionov, A., Choo, H.: Agent-based simulation system AGNES for networks modeling: review and researching. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2012), Paper 115. ACM (2012)
15. Gangemi, A., Presutti, V.: Ontology design patterns. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. IHIS, pp. 221–243. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92673-3_10

16. Zagorulko, Y., Borovikova, O., Zagorulko, G.: Development of ontologies of scientific subject domains using ontology design patterns. In: Kalinichenko, L., Manolopoulos, Y., Malkov, O., Skvortsov, N., Stupnikov, S., Sukhomlin, V. (eds.) DAMDID/RCDL 2017. CCIS, vol. 822, pp. 141–156. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96553-6_11
17. Antoniou, G., van Harmelen, F.: Web ontology language: OWL. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. INFOSYS, pp. 67–92. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_4
18. Protégé. https://protege.stanford.edu. Accessed 05 May 2019
19. SWRL: A semantic web rule language combining OWL and RuleML. http://www.w3.org/Submission/SWRL/. Accessed 05 May 2019