THE ARCHITECTURE OF SYSTEM FOR ANALYSIS AND OPTIMIZATION OF PARALLEL PROGRAMS

Aleksej Otwagin[1)], Rauf Sadykhov[1)], Aleksej Bogomazov[2)]

[1)] Belarusian Academy of Sciences, United Institute of Informatics Problems (UIIP),6 Sourganov str., Minsk, 220012, Belarus. e-mail: *forlelik@yahoo.com*.

[2)] Belarusian State University of Informatics and Radioelectronics, 6, P.Brovka str., Minsk, 220027, Belarus.

Abstract: The architecture of system for optimization of parallel programs that allows the development, optimization and preparing of parallel programs is offered. The architecture of system is constructed as open system that it could be adjusted easily for use of various support libraries for parallel programs and optimization techniques.

Introduction

The contemporary technologies of computer information processing require the wide application of highly productive systems with the support of parallel computations. The parallel computing system (PCS) to solve the search can be implemented by two ways:

1. At first the hardware is projected and then the software is created. The hardware usually represents the specialized architectural components distinguished both high efficiency and high cost. The programs to be developed for such systems are closely adhered to its architecture, therefore at the further development and replacement of the equipment of them it is necessary to upgrade, and sometimes to create fully new systems.
2. The unified libraries to support the virtual program model providing an intermediate layer between libraries of operational system and the programming environment of PCS on the one hand, and physical architecture of PCS with another hand are developed. The porting of virtual program model on other physical architecture requires only small changes in already developed software. Such approach not only saves resources, but also conducts to evolutionary development of complex software. The computer itself can be grouped from rather cheap components, which essentially reduce cost of system, providing productivity that is comparable with specialized architectures.

The methodology for designing of parallel algorithms [3], that substantially ensures protection of designer from potential mistakes, assumes parallel development which considers machine-independent aspects of algorithm implementation, such, as parallelism, at the first stage, and the features of designing, that are connected with concrete PCS architecture on the second one. The given approach divides a process of designing on four separate stages: partitioning, communications, agglomeration and mapping. All these stages can be in part or fully-automated. There is a necessity to create the automation tools which would allow developers of parallel programs to pay the basic attention to own algorithms of information processing, instead of to the decision of questions for optimum distribution of subtasks in the computing system. Besides automation can essentially facilitate the procedure of correct description of the parallel program in view of requirements of used support library of virtual program model.

The common problems in development of parallel applications

The majority of the existing parallel applications developed for multi-purpose parallel systems, contain two different types of functions:

- Functions, which directly realizes certain algorithm of information processing;
- Functions of adaptation of algorithm to PCS architecture.

Such division of the application code determines some important features of its designing within the framework of existing languages and tools of parallel programming, namely:

- The parallel application is more difficult for developing at use of features of hardware architecture;
- The received algorithm is not portable even for insignificant distinguished architectures;
- It is difficult enough to select a machine-dependent initial code from the general algorithm, which is automatically optimized to guarantee mobility and productivity.

A tool for development of parallel programs should contain two interconnected components, namely [1]:

- high level language of parallel programming with the obvious parallelism containing precisely determined basic primitives for parallel programs;

- the set of tools for compilation and optimization, taking into account all the aspects of construction of the program, and allowing to achieve the maximal degree of productivity, mobility and support for parallel software.

There are many tools for automation of the parallel programming, constructed as the integrated environments for visual development. As example it is possible to serve the software packages CODE [9], HeNCE, GRADE [7], EDPEPPS [2], etc. Usually such tools use standardized libraries of parallel computing support as a basis for construction of parallel programs. At present most widely used libraries are MPI (Message Passing Interface) [6] (implementations MPICH [5], LAM/MPI) or PVM (Parallel Virtual Machines) [4].

A complete tool for automation of parallel programs must contain components for the automation of the following problems:

- Observance of designing methodology of programs;
- Maintenance of architecture-independent development;
- Support of various libraries of parallel computing;
- Presence of optimization procedures and adjustment of programs for the maximal productivity.

Architecture of system for analysis and optimization of parallel programs

The proposed architecture of system for analysis and optimization of parallel programs is shown on fig. 1.

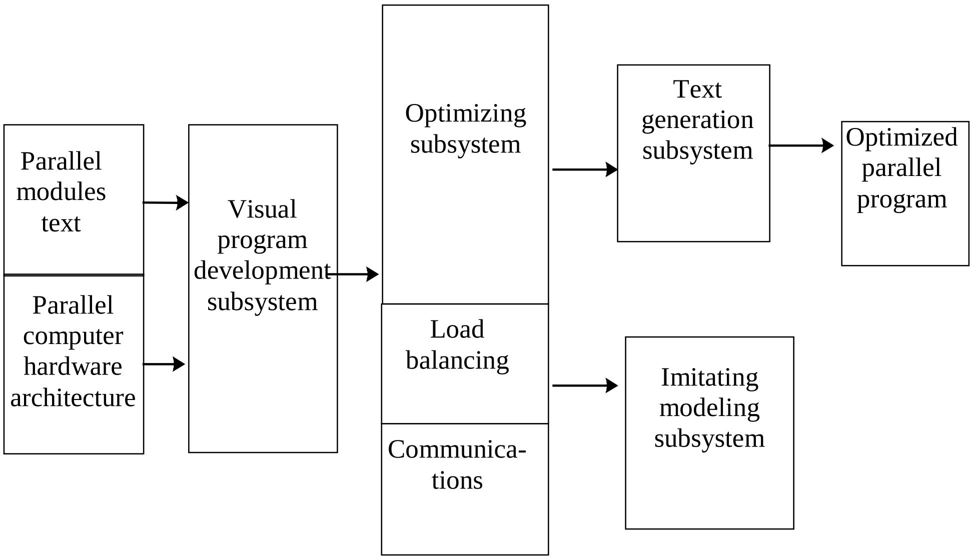
Parallel modules text
Parallel computer hardware architecture
Visual program development subsystem
Optimizing subsystem
Load balancing
Communica-tions
Text generation subsystem
Optimized parallel program
Imitating modeling subsystem

Fig. 1. The architecture of system for parallel program development.

The structure of the suggested architecture of system includes the following subsystems:

- visual program development;
- optimization of programs taking into account of hardware architecture;
- imitating modeling of system functioning;
- generation of a skeleton of the parallel application.

The subsystem of visual development is intended for graphic input and editing of parallel programs structures and topology of a parallel computer. Usually parallel algorithms and PCS architecture are represented as graphs. During editing of this graphs the characteristics of separate elements are defined. For modules of the parallel program the computing complexity, requirements to the specialized resources and, probably, other characteristics, for example directive time for completion (for real-time systems) are determined. For the processing modules included in physical architecture of parallel system, productivity and presence of the specialized resources are defined. The created information models can be exported to XML files. During creation of models there is transformation to object-oriented representation for the subsequent analysis.

The analysis tools are applied first of all on optimization of productivity of the parallel program. Time of running of the program, efficiency of use of the equipment, scalability and cost of development enters into concept of productivity.

The subsystem of optimization of the parallel program solves the distribution problems of set of branches and modules of the parallel program onto processors of the computing system, and also a problem of optimization of intra-program communications. For the solution of a problem of modules distribution the algorithm of decomposition of the parallel program, based on model of a virtual network [12, 13] is used. The given algorithm allows achieving the well-balanced loading of processors.

During optimization the algorithm forms clusters, minimize general time of program execution and information exchanges. The algorithm can be used both for static and dynamic models, and is easily scaled on any number of tasks and processors. Besides testing of algorithm for various models of communication networks (with fixed and varied time of the message transfer) its efficiency is testified.

Use in model of a virtual network the methods of the theory of genetic algorithms [8] in a combination to technologies of training of neural networks allows carrying out continuous optimization during search of decisions. In comparison with other optimization algorithms, for example with classical genetic algorithm, the algorithm of a virtual network shows the best productivity at increase in complexity of the parallel program. Besides the algorithm of a virtual network, alongside with optimization of time of parallel program execution increases uniformity of loading of processors and promotes more to an effective utilization of the computing system.

Problems of effective realization of information exchanges inside the parallel program are closely connected to a problem of distribution of the program modules onto processors. At an exchange of messages between processors the delays can occurs because of features of algorithm realization (for example, at the moment of acceptance of the message by the processor it is not sent yet by other processor), and because of the hardware reasons (for example, network congestions). At designing stage it is necessary to reduce to a minimum of the expenses for communications and to make the relation of time of calculations T_{op} to time of communications T_{comm} so large as possible. For maintenance of reliable interaction within the framework of parallel system it is required to provide effective routing of messages in a communication subsystem. The routing should provide the maximal speed of transfer without overloads of communication links. Overloads are expressed in the excessive volumes of the information transmitted on separate routes owing to what there are delays in performance of branches of parallel algorithm. Many routing algorithms do not take into account this fact.

For ensuring the uniformity of loading on the communication links with simultaneous maximizing of transfer speed the method of the static wormhole routing based on technology of genetic algorithms [11] is used. The given approach allows achieving the optimization of the routing table with various criteria for short time as assumes parallel search in space of possible solutions. Alongside with optimization of routes it is possible to determine the potential bottlenecks arising because of interdependence of message transfers, and to remove this lacks at designing set of routes.

After optimization procedures the model of parallel program can be investigated by methods of imitating modeling. Imitating modeling allows receiving the information about performance of separate modules, the moments of information interaction, and to estimate the time of execution of the parallel program, and also other characteristics. On the basis of these data the developer can make the decision on change of the program, for example association of some modules in larger, or, on the contrary, decomposition the module on a little bit fine. After modification in information model the process of the analysis can be repeated.

Results of the analysis are used at generation of the text of the parallel program. Automation of generation of the text allows the programmer to concentrate maximum on algorithm of processing of the information. Thus features of virtual program model can be hidden from the user. During generation the skeleton of the application containing calls of the information processing procedures and communication exchanges is created. The skeleton of the application can be recompiled in the program for the certain library of parallel computing support. For this purpose the optimized model of the program can be represented as some parallel pseudo-code, which will be transformed into the concrete program at use of a subsystem of translation.

Conclusion

The proposed architecture for system to support the design of parallel programs allows creating the tools of automation providing essential acceleration of program development process. The specified tools can carry out automation of the control of the parallel program correctness, hide from the developer machine-dependent aspects of the organization of the application, using the various libraries of parallel computing support. Architecture modularity and openness assumes simple opportunities of its updating and expansion.

At present the development of a prototype of system, constructed on the resulted architecture is ended.

References

[1]. Baiardi F., Danelutto M., Jazayeri M. et al. Architectural models and design methodologies for general-purpose highly-parallel computers. // Proceedings of the IEEE CompEuro '91 – Advanced Computer Technology, Reliable Systems and Applications. – May 1991. – Bologna, Italy. – p. 18 – 25.

[2]. Delaitre T., Ribeiro Justo G.R. et al. EDPEPPS: An environment for the design and performance evaluation of portable parallel software. // Euromicro Workshop PDP'97. – London. – June 1997. – p. 401 – 406.

[3]. Foster Ian. Designing and Building Parallel Programs. Addison-Wesley, 1995. – 430 p.

[4]. Geist A., Beguelin A., Dongarra J. et al. PVM: A Users' Guide and Tutorial for Networked Parallel Computing. – MIT Press, 1994. – 299 p.

[5]. Gropp W., Lusk E. Installation and user's guide for mpich, a portable implementation of MPI. – Technical Report ANL-01/x. – Argonne National Laboratory, 2001. – 179 p.

[6]. Gropp W., Lusk E., and Skjellum A. Using MPI: Portable Parallel Programming with the Message Passing Interface. – MIT Press, 1995. – 350 p.

[7]. Kacsuk P., Dózsa G., Lovas R. The GRADE Graphical Parallel Programming Environment. // Parallel Program Development for Cluster Computing, Methodology, Tools and Integrated Environments. – Nova Science Publishers, Inc. – 2001. – p. 231-247.

[8]. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, 1994. – 340 p.

[9]. Newton P., Browne J.C. The CODE 2.0 Graphical Parallel Programming Language. // Proc. ACM Int. Conf. on Supercomputing. – July 1992. – p. 167 – 177.

[10]. Ni L. McKinley P. A survey of wormhole routing techniques in direct networks. // Computer. – February 1993. – p. 62 – 76.

[11]. R.Kh. Sadykhov, A.V. Otwagin, V.A. Radishevsky. The algorithm of statical message routing in real-time systems on the basis of genetic algorithms // The BSTU bulletin. Engineering, automation, and computers. - 2001. - 4 (10). - p. 97-99. (In Russian).

[12]. R.Kh. Sadykhov, A.V. Otwagin. Solution search algorithm of solution search for systems of parallel processing based on a virtual neural network model. // Avtomatika I vychislitel’naya tekhnika. - №1. – 2001. Riga, Latvia, p. 25-33. (In Russian).

[13]. Yufik Y. M., Sheridan T. B. Virtual Networks: New framework for operator modeling and interface optimization in complex supervisory control systems // A Rev. Control. – V. 20. – p. 179-195.