# Some Examples of Parallel Programs Based on DVM approach

Leonid P. Kamenshchikov*

Institute of Computational Modelling of Siberian Branch of RAS,
Krasnoyarsk, 660036, Russia

**Key words**: parallel computing, Fortran-DVM.

Parallel computing on rather cheap clusters has been gaining more and more distribution among broad range of programmers in recent years [1]. Usage of well-known MPI-approach for parallel programming has significant disadvantages, in particular, [2]: the development and debugging of such programs requires much more effort from the programmer because the level of language is too low; the efficient program execution on clusters requires load balancing which is difficult to provide in MPI-approach, etc.

DVM-system developed in Keldysh Institute of Applied Mathematics of Russian Academy of Sciences allows to create parallel programs in C-DVM and Fortran-DVM languages for different architecture computers and computer GRIDs [3--5]. (For brevity we shall speak only about a Fortran below).

Fortran-DVM is a set of extensions to the Fortran 77 standard that permits programmers to distribute data among multiple processors. Using Fortran-DVM languages a programmer deals with the only one version of the program both for serial and parallel execution. Besides algorithm description by means of Fortran 77 features the program contains rules for parallel execution of the algorithm. These rules are syntactically organized in such a manner that they are "invisible" for standard Fortran compilers and doesn't prevent DVM-program execution and debugging on personal computers or workstations as usual serial program.

Following features of parallel program execution specification is provided to a programmer [4]: distribution of array elements over the processors; distribution of cycle iterations over the processors; specification of parallel executed program sections (parallel tasks) and their mapping on the processors; organization of efficient access to remote data (located on other processors); organization of efficient execution of reduction operations, which are global operations on the data located on different processors (such as summing values or finding their maximal and minimal values). The compiler translates Fortran-DVM program in the program on standard Fortran language, expanded by DVM-program runtime support system functions. Runtime system uses standard communication libraries (MPI, PVM, Router) for interprocessor communications.

The main goals of the DVM-system are follows [4]. Simplicity of parallel program development. Portability of parallel program onto different architecture computers (serial and parallel). For serial computers the portability is provided by DVM-directive "transparency" for standard Fortran 77 compilers. High performance of program execution. Reusability (composition of parallel applications from several modules). Unified parallelism model for Fortran 77 languages, and, as result, unified system of runtime support, debugging, performance analyzing and prediction.

The program debugging is performed as follows [4,6]. At first step, the program is debugged at a workstation as a usual sequential program using ordinary debugging tools. At the second step the program is executed at the same workstation using special mode to check DVM-directives. At the third step the program may be executed at parallel computer (or its model, for example, MPI-machine in Windows 9.x or Linux environment) in the special mode when intermediate results of parallel execution are compared with the reference ones (for example, the re-

* E-mail: lk@icm.krasn.ru

sults of its sequential execution). To debug the program on real parallel computer the trace accumulation tools are used.

In this report the DVM-approach is applied to three test problems.

1) Computing of $\pi$ by integrating function $f(x)= 4/(1+x^2)$ on interval [0,1]. The comparison of efficiency for DVM and MPI approaches is given.

2) Solving of 3-D Possion's equation with Dirichlet boundary condition by multigrid method [8].

3) Solving of 3-D transient systems of partial differential equations for modelling of reaction-diffusion catalytic non-isothermal reactions both on surface of the catalyst and inside of it. Such systems driven far from equilibrium by mass or energy fluxes may develop space or time symmetry breaking bifurcations that lead to time dependent oscillations or space periodic structures.

As a parallel computing platform we have employed PC-clusters MBC-1000/16 and MBC-1000M [9].

**References**


1. V.V. Voevodin, Vl.V. Voevodin. Parallel Computations. BHV-Petersburg. Saint Petersburg, 2002 (in Russian).
2. V. Arzhanov, A. Voronkov, A. Golubev, E. Zemskov, N. Konovalov, V. Krukov. "Development of Explicit Cyclic Schemes with Chebyshev's Polynomials for Space Neutron Kinetics on the Multiprocessor Computers", Proceedings of INFORUM-99 Annual Meeting on Nuclear Technology, pp. 19--23, Karlsruhe, 1999.
3. N.A. Konovalov, V.A. Krukov, S.N. Mihailov and A.A. Pogrebtsov. "Fortran DVM - a Language for Portable Parallel Program Development", Proceedings of Software For Multiprocessors & Supercomputers: Theory, Practice, Experience, pp. 124--133, Institute for System Programming of RAS, Moscow, 1994.
4. DVM-System. http://www.keldysh.ru/dvm
5. N.A. Konovalov, V.A. Krukov, S.N. Mikhailov, and A.A. Pogrebtsov. Fortran DVM - A Language for the Development of Portable Parallel Programs. Programming and Computer Software, v 21, \No\ 1, 1995, pp. 35--38.
6. V.A. Krukov, R.V. Udovichenko. Debugging DVM Programs. Programming and Computer Software, v 27, \No\ 3, 2001, pp. 123--129.
7. V.A. Krukov. Development of Parallel Programs for Computing Clusters and GRIDs. http://www.keldysh.ru/pages/dvm/dvmhtm1107/publishr/clust-141201.htm
8. V.V. Shaidurov. Multigrid Methods for Finite Elements. Kluwer Academic Publishers. 1995.
9. MBC-1000M. http://www.jscc.ru