

## DVM SYSTEM FOR PARALLEL PROGRAMMING

**V. Bakhtin<sup>1</sup>, A. Kolganov<sup>2</sup>, V. Krukov<sup>3</sup>, N. Podderyugina<sup>4</sup>, M. Pritula<sup>5</sup>**

<sup>1</sup> *KIAM RAS, MSU, Moscow, [bakhtin@keldysh.ru](mailto:bakhtin@keldysh.ru)*

<sup>2</sup> *KIAM RAS, MSU, Moscow, [79854210702@ya.ru](mailto:79854210702@ya.ru)*

<sup>3</sup> *KIAM RAS, MSU, Moscow, [krukov@keldysh.ru](mailto:krukov@keldysh.ru)*

<sup>4</sup> *KIAM RAS, Moscow, [konov@keldysh.ru](mailto:konov@keldysh.ru)*

<sup>5</sup> *KIAM RAS, Moscow, [pritula@keldysh.ru](mailto:pritula@keldysh.ru)*

DVM-system[1] was developed in Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, with the active participation of graduate students and students of Faculty of Computational Mathematics and Cybernetics of Lomonosov Moscow State University. It is designed to create parallel programs of scientific-technical calculations in C-DVMH and Fortran-DVMH languages. These languages use the same model of parallel programming (DVMH-model) and are the extensions of standard C and Fortran languages by parallelism specifications, implemented as compiler directives. The directives are invisible to standard compilers, so a programmer can have one program for sequential and for parallel execution on computers of different architectures.

DVMH-model allows to create efficient parallel programs (DVMH-programs) for heterogeneous computational clusters, which nodes use as computing devices not only universal multi-core processors but also can use attached accelerators (GPUs or Intel Xeon Phi coprocessors).

C-DVMH and Fortran-DVMH compilers convert the source program into a parallel program using standard programming technologies MPI, OpenMP and CUDA.

DVM-system includes the tools of functional debugging and performance debugging of DVMH-programs.

The report presents the experience of parallelizing various application programs using DVM-system.

### References

1. <http://dvm-system.org/>