Part II
Programming

In Part II is devoted to programming of parallel computers. It is designed in a way that every reader can exploit the parallelism of its own computer, either on multi-cores with shared memory, or on a set of interconnected computers, or on a graphic processing units. The knowledge and experience obtained can be beneficial in eventual further, more advanced applications, which will run on many-core computers, computing clusters, or heterogeneous computers with computing accelerators.

We start in Chap. 3 with multi-core and shared memory multiprocessors, which is the architecture of almost all contemporary computers, that are likely the easiest to program with an adequate methodology. The programming such systems is introduced using OpenMP, a widely used and ever expanding application programming interface well suited for the implementation of multithreaded programs. It is shown how the combination of properly designed compiler directives and library functions can provide a programming environment where the programmer can focus mostly on the program and algorithms and less on the details of the underlying computer architecture. A lot of practical examples are provided, which help the reader to understand the basic principles and to get a further motivation for fully exploiting available computing resources.

Next, in Chap. 4, distributed memory computers are considered. They cannot communicate through the shared memory therefore messages are used for the coordination of parallel tasks that run on geographically distributed but interconnected processors. Definition of processes with their management and communication are well defined by a platform-independent message passing interface (MPI) specification. The MPI library is introduced from the practical point of view, with basic set of operations that enable the implementation of parallel programs. Simple example programs should serve as an aid for a smooth start of using MPI and as motivation for developing more complex applications.

Finally, in Chap. 5, we provide an introduction to the concepts of massively parallel programming on GPUs and heterogeneous systems. Almost all contemporary desktop computers are multi-core processor with a GPU units. Thus we need a programming environment in which a programmer can write programs and run them on either a GPU, or on a multi-core CPU, or on both. Again, several practical examples are given that help and assist the readers in acquiring knowledge and experience in programming GPUs, using OPenCL environment.