Part III
Engineering

The aim of Part III is to explain why a parallel program can be more or less efficient. A basic approaches are described for the performance evaluation and analysis of parallel programs. Instead of analyzing complex applications, we focus on two simple cases, i.e. a parallel **computation of number** π, by using numerical integration, and a solution of simplified **partial differential equation** on 1-D domain, by using explicit solution methodology. Both cases, already mentioned in previous chapters, even so simple, they already incorporate most of possible pitfalls that could arise during their parallelization. The first case, computation of *pi*, requires just a few communication among parallel tasks, while in the explicit solution of PDE, each process communicate with its neighbors in every time step.

Besides these two cases, we also evaluate the **Seam Carving** algorithm in terms of performance on CPU and a GPU platform. Seam Carving is an image processing algorithm in 2-D domain and as such appropriate for implementation on GPU platforms. It comprises a few steps of which some cannot be effectively parallelized.

Parallel programs run on adequate platforms, i.e. multi-core computers, interconnected computers or computing clusters, and GPU accelerators. After an implementation of any parallel program, several questions remain to be answered, e.g.:

- How the execution time decreases with larger number of processors?
- How many processors are optimal for a specific task?
- Will execution time always decrease, if the number of processors is increased?
- Which parallelization methodology provides the best results?
- and similar.

We will answer the above questions by running the programs with different parameters, e.g. size of the computation domain and the number of processors. We will follow also the execution efficiency and limitations that are specific for each of the three parallel methodologies: OpenMP, MPI, and OpenCL.

An electronic extension of the Engineering part will be permanently available on a book web, hosted by Springer server. Our aim is that it become a vivid forum of readers, students, teachers and other developers. We expect your inputs in a form of your own cases, solutions, comments, and proposals. Soon after the publication of this book more complex cases will be provided, i.e. a numerical solution of a 2-D diffusion equation, a simulation of N-body interactions with possible application in molecular dynamics, and similar.

Each of engineering cases will be introduced with a basic description of the selected problem, sequential algorithm and its solution methodology. Then, for all considered parallelization approaches: OpenMP, MPI and OpenCL, initial parallel algorithms will be developed and their expected performance will be estimated. Results will be compared in terms of programming complexity, execution time, and scalability. The complete implementations will be provided with an adequate program code. Any improvements and feed back from all users are welcome.