Chapter 4
Algebra-Dynamic Models for CPU- and GPU-Parallel Program Design and the Model of Auto-Tuning

ABSTRACT

This chapter considers algebra-dynamic models of parallel programs, which are based on concepts of transition systems theory and algebra of algorithms. The models of sequential and parallel multithreaded programs for multicore processors and program models for graphics processing units are constructed. The authors describe transformations of programs aimed at transition from sequential to parallel versions (parallelization) and improving performance of parallel programs in respect to execution time (optimization). The transformations are based on using rewriting rules technique. The formal model of program auto-tuning as an evolutional extension of transition systems is proposed, and some properties of programs are considered.

INTRODUCTION

Development of multicore processors leads to increasing importance of parallel programming aimed at standard, widely accessible computers, and not just for specialized high-performance systems (Akhter & Roberts, 2006). However, there is one more direction of parallel programming which

DOI: 10.4018/978-1-5225-9384-3.ch004

has received especial development recently, namely, the programming of general-purpose tasks for graphics processing units (GPUs) ("General-Purpose Computation on Graphics Hardware", n.d.). Market requirements have led to rapid development of GPUs and, at present, their computing capacity considerably exceeds the capabilities of usual processors. Therefore, GPUs were applied for solving the problems not concerned directly with graphics processing. Research in these directions is supported by GPU developers: in particular, NVIDIA company provides CUDA platform for general-purpose computations on GPUs ("NVIDIA CUDA technology", n.d.).

Despite the presence of specialized facilities for CUDA, development of GPU programs remains a labor-consuming work, which requires from a developer a knowledge about low-level details of hardware and software platform. Therefore, there is a need of research in the area of automation of software development process for GPUs. This chapter describes the development of formal design methods, based on concepts of transition systems theory, algebraic programming and algebra-dynamic models of programs (Andon, Doroshenko, Tseytlin, & Yatsenko, 2007; Doroshenko, Zhereb, & Yatsenko, 2010) with the use of rewriting rules technique (Doroshenko & Shevchenko, 2006; "TermWare", n.d.) for automated development of efficient programs for GPUs. High-level models of programs and models of program execution are developed for central processing unit (CPU) and GPU. Application of rewriting rules and high-level models for automated parallelization and optimization of programs for GPUs is described. The method of automated transition between a high-level model of a program and a source code, which is based on the use of special rewriting rules is proposed. This chapter also considers the formal model of program auto-tuning constructed as an evolutional extension of transition systems and properties of programs.

Currently there is a significant amount of research in the area of automation of software development for graphics processors. Research community examines problems of transition from sequential to parallel programs as well as problems of optimization of existing parallel programs with the use of GPUs. Particularly, the paper (Lee, Min, & Eigenmann, 2009) considers the automatic transition from a multithreaded program implemented using OpenMP technology ("OpenMP Application Programming Interface", 2015) to implementation of the same program on CUDA platform. Paper (Baskaran et al., 2008) describes a platform for loop optimization in GPU programs. Systems for automatic parallelization and optimization of programs from a

Related Content