

Programming Language for Hierarchical, Many-Core, Parallel Architectures using ManyDSL

Piotr Danilewski

“Languages shape thought” – for that reason it is important which language we choose for programming parallel processors. The many-core processors exist on the market for over a decade, yet there is no language that would reflect their unique traits and capabilities well. Popular languages, such as CUDA, are merely extensions to existing, old languages such as C, which have different assumptions about the architecture, that do not match well to parallel processors.

The goal of this project is to fill this void – to create a new language from scratch, with many-core programming in mind. In particular, we want to draw attention to memory hierarchy, splitting work between the threads, address the inter-thread communication and reduce chance for the user to make errors caused by asynchronous execution.

As part of the project we are going to present an abstract description of the language in a form of grammar rules. We are also going to create an actual compiler, using the ManyDSL tool. The ManyDSL tool provides an innovative approach of defining languages which we have worked on for the last few years.

We are going to evaluate the language with respect to efficiency of the produced machine code, as well as ease of programming. Typical problems in the domain should find an expressive representation in the language, increasing the productivity of the programmer. We expect that a new way of thinking induced by our language will help better understand the unique characteristics of the many-core architecture.