

Autor: Dalvan Jair Griebler

Título: Domain-Specific Language & Support Tools for High-Level Stream Parallelism.

Ingresso: 2012/1

Defesa: 30/03/2016, ata nr. 62 **Homologação:** xx/xx/xxxx ata nr./

Linha de Pesquisa: Processamento Paralelo e Distribuído.

Orientador: Prof. Dr. Luiz Gustavo Leão Fernandes

Orientador de Cotutela: Dr. Marco Danelutto (Università di Pisa)

Comissão Examinadora: Prof. Dr. César Augusto FonticIELha De Rose (PPGCC/PUCRS), Dr. Marco Aldinucci (University of Torino) e Dr. José Daniel García Sánchez (Universidad Carlos III de Madrid).

Abstract:

Stream-based systems are representative of several different application domains including video, networking, audio, graphic processing, etc. Stream parallel programs may run on different kinds of parallel architectures (desktop, servers, cell phones, and supercomputers) and represent significant workloads on our current computing systems. Nevertheless, most of them are still not parallelized. Moreover, when a new one has to be developed, programmers have to face a trade-off between coding productivity, coding portability, and performance. To solve this problem, we provide a new Domain-Specific Language (DSL) that naturally/on-the-fly captures and represents parallelism for stream-based applications. The aim is to offer a set of attributes (through annotations) that preserves the source code of the program and is not architecture-dependent. We used the C++ attribute mechanism to design a "de-facto" standard C++ embedded DSL named as SPar. However, the implementation of DSLs using compiler-based tools is difficult, complicated and usually requires a significant learning curve. This is even harder for those that are not familiar with compiler technology. Our motivation is, therefore, to simplify this path for other researchers (experts in their domain) with support tools (our tool is CINCLE) to implement high-level and productive DSLs through powerful and aggressive source-to-source transformations. In fact, parallel programmers can use their expertise without having to design and implement the low-level code. The principal goal of the thesis was to create a DSL and support tools for high-level stream parallelism in the context of a programming framework that is compiler-based and domain-oriented. Thus, we

implemented SPar using CINCLE. SPar supports the software developer with productivity, performance and code portability while CINCLE provides sufficient support to generate new DSLs.

Also, SPar targets source-to-source transformation producing parallel pattern code implemented on top of FastFlow and MPI. Finally, we have a full set of experiments that shows that SPar provides better coding productivity without significant performance degradation in multi-core systems as well as transformation rules that are able to achieve code portability through its attributes.