A Skeleton Library

Herbert Kuchen

University of Münster, Department of Information Systems
Leonardo Campus 3, D-48159 Münster, Germany

kuchen@uni-muenster.de

Abstract

Today, parallel programming is dominated by message passing libraries such as MPI. Algorithmic skeletons intend to simplify parallel programming by increasing the expressive power. The idea is to offer typical parallel programming patterns as polymorphic higher-order functions which are efficiently implemented in parallel. Ideally, a parallel application can be generated by combining these skeletons. Among others, communication problems like deadlocks and starvation are avoided and a cost model allows to predict the performance of an application. Unfortunately, existing skeleton systems differ considerably. Moreover, they are typically offered in form of a new programming language, which parallel programmers may hesitate to learn. Both aspects have so far prevented the practical use of skeletons. The approach presented here integrates the main features of existing skeleton systems. Moreover, it does not come along with a new programming language or language extension, which parallel programmers may hesitate to learn, but it is offered in form of a library, which can easily be used by e.g. C and C++ programmers. A major technical difficulty is to simulate the main requirements for a skeleton implementation, namely higher-order functions, partial applications, and polymorphism as efficiently as possible in an imperative programming language. Experimental results based on a draft implementation of the suggested skeleton library show that this can be achieved without a significant performance penalty. For details see [Ku02].

References

[Ku02] H. Kuchen: The Skeleton Library Web Pages,
http://danae.uni-muenster.de/lehre/kuchen/Skeletons/