HaskellMPI
Entwicklung paralleler Programme in Haskell

Michael Weber
Lehrstuhl für Informatik II
RWTH Aachen

michael.weber@i2.informatik.rwth-aachen.de

Funktionale Programmiersprachen wie ML und Haskell erlauben durch ihren hohen Abstraktionslevel sehr elegante und kurze Beschreibungen vieler Algorithmen. Durch die Syntax, die eng an mathematische Schreibweisen angelehnt ist, wird ferner die Prüfung auf Übereinstimmung einer Implementierung mit der tatsächlichen formalen Spezifikation unterstützt. Daher eignen sich funktionale Programmiersprachen insbesondere für einen *Rapid Prototyping*-Ansatz bei der Entwicklung von Algorithmen.

Aufgrund der steigenden Größe vieler Problemstellungen hat sich in den letzten Jahren die Parallelisierung von Algorithmen als wichtiger Forschungsbereich etabliert. Es ist also naheliegend, über Möglichkeiten zur Formulierung von parallelen Algorithmen in funktionalen Sprachen nachzudenken, um auch dort deren Vorteile ausnutzen zu können. Für Haskell und verwandte Programmiersprachen existieren einige Ansätze wie *Glasgow Parallel Haskell*, *Glasgow Distributed Haskell* sowie der *Eden*-Compiler, durch die Operationen zur parallelen Programmierung bereitgestellt werden. Programme, die in diesen Sprachen entwickelt werden, erfordern jedoch ein spezielles Laufzeitsystem, das vom jeweiligen Compiler eingebunden wird. Weiterhin sind diese Programme aufgrund syntaktischer Unterschiede der Sprachen nicht zwischen verschiedenen Compiler-Systemen austauschbar.

In diesem Vortrag wird mit HaskellMPI eine Bibliothek vorgestellt, mit der man in Standard-Haskell die Möglichkeit erhält, parallele Programme zu schreiben. Diese sind nicht abhängig von speziellen Haskell-Implementierungen, so daß zur Unterstützung eines weiteren Compilers lediglich die HaskellMPI-Bibliothek portiert werden muß. Ferner werden einige Beispiele für die Anwendung der Bibliothek gegeben.