

BSc Arbeit: Implementierung von Performance Assertions für MPI Metriken

Die Effizienz und Leistung von Software ist von vielen Faktoren im Wechselspiel zwischen Algorithmus, Rechnerarchitektur Programmiersprache und Middleware abhängig. Eine Methode zur Erfassung und Überprüfung der Leistung sind sogenannte *Performance Assertions*^[1]. Mit Hilfe dieser Assertions werden in der zu überprüfenden Anwendung leistungsrelevante Bereiche markiert und mithilfe von Ausdrücken die zu erwartende Leistung formuliert. Zur Laufzeit werden diese Assertions mittels einer geeigneten Messumgebung auf Korrektheit überprüft. Schlägt eine solche Assertion fehl, gibt es mehrere Möglichkeiten der Reaktion, angefangen von einer Fehlermeldung bis hin zu einer Änderung des Programmverhaltens. Aktuell existiert jedoch keine allgemein verfügbare Umsetzung dieses Konzeptes, insbesondere auch nicht für die gängigen Programmiersprachen C/C++ oder FORTRAN.

Im Rahmen dieser Arbeit sollen für die Hochsprache C++ auf MPI spezialisierte Performance-Assertions zur Laufzeitüberprüfung von MPI Anwendungen implementiert werden. Diese Assertions sollen für einen definierten Code-Block folgende Metriken ermöglichen, wie sie aus der MPI Analyse bekannt sind:

- Wartezeit an MPI Konstrukten
- das Verhältnis von Kommunikation und Berechnung
- den Overlap-Faktor, d.h. das Verhältnis von der gesamten Laufzeit des Blocks zur Summe von Rechenzeit und der real benötigten Kommunikationszeit, incl. der durch asynchrone Kommunikation verborgenen Kommunikationszeit.
- Hardwarezähler basierte Metriken, wie z.B. die Anzahl der Fließkommaberechnungen, die durch die Hardware ausgeführt wurden.

Diese Metriken sollen mittels einfacher aus dem Programm übernommener Variablen und Konstanten parametrisierbar sein, um unterschiedlicher Rechenlasten und Architekturen Rechnung zu tragen. Beispiele für solche Variablen sind die Anzahl der MPI Prozesse, die Anzahl der Elemente, die verarbeitet werden sollen und die Laufzeit der Anwendung.

Ziel der Arbeit ist es, ein auf dem ROSE Compiler Framework basierendes prototypisches Werkzeug zu entwickeln, das die in den Ziel-Quellcode vorhandenen und durch Pragmas definierten Assertions in ausführbaren Code transformiert und mittels eines geeigneten Messsystems zur Ausführungszeit überprüft.

Voraussetzungen sind

- grundlegende Kenntnisse der Compilertechnologie und der MPI Programmierung (oder die Bereitschaft, sich diese selbständig zu erarbeiten)
- gute Kenntnisse der C / C++ Programmiersprachen

Im Rahmen dieser Arbeit werden Sie

- sich Wissen erarbeiten über
 - Programmtransformationen,
 - verschiedene Performance Analyse Werkzeuge und
 - Methoden zur Erfassung von Leistungscharakteristika aktueller Prozessorensowie
- eine Bachelor-Thesis im Umfang von ca. 30-40 Seiten verfassen.

[1] Jeffrey S. Vetter and Patrick H. Worley. 2002. Asserting performance expectations. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing* (Supercomputing '02). IEEE Computer Society Press, Los Alamitos, CA, USA, 1-13.