

Diplom- / Masterarbeit

Datum: 30. August 2010

Themenbereich: Testfallgenerierung zur Unterstützung der Anwendungsparallelisierung

Ausgangssituation:

Das Parallelisieren sequentieller Anwendungen stellt Softwareentwickler vor eine sehr schwierige Aufgabe:

Wie erreiche ich mit möglichst wenig Änderungen am vorliegenden sequentiellen Code eine möglichst hohe Beschleunigung (engl. speed up)?

Konkret bedeutet das, es müssen zunächst mögliche Ansatzpunkte zur Parallelisierung im Code gefunden werden, ehe überprüft werden kann, ob die Daten an den identifizierten Stellen für die gleichzeitige Verwendung in mehreren Ausführungsfäden geeignet sind. Ist dies nicht der Fall, muss entschieden werden, mit welchem Synchronisationskonstrukt eine vorhersagbare und korrekte Ausführung erreicht werden kann. Wettläufe oder Verklemmungen wären sonst die Folge. Als wäre das nicht schon kompliziert genug muss alles so realisiert werden, dass immernoch eine Beschleunigung erreicht wird, die groß genug ist, um den Aufwand zu rechtfertigen - ansonsten ist die Parallelisierung nutzlos.

Ein häufig auftretendes Muster bei parallelen Anwendungen ist, mehrere Instanzen eines Algorithmus gleichzeitig an denselben Daten arbeiten zu lassen. Es muss hierbei nun nicht nur erkannt werden, auf welchen Daten Wettläufe stattfinden können, sondern auch, welche Art der Synchronisierung zu deren Vermeidung anzuwenden ist (Fadenlokale Kopie der gemeinsamen Daten, partieller Zugriff, sequentieller Zugriff,...). Diese Entscheidung muss bisher vom Entwickler selbst getroffen und validiert werden, was zeitaufwändig ist und darüberhinaus natürlich fehleranfällig.

Ziel:

Ziel dieser Arbeit ist es, Testfälle zu generieren, die verschiedene Zugriffsmuster auf eine zu parallelisierende Codestelle anwenden um das einzusetzende Synchronisationskonstrukt zu bestimmen. Die Testfälle sollen daher folgende Ergebnisse liefern:

1. Korrektheit
 - Entspricht das Ergebnis dem vor der Ausführung?
 - Wenn nicht, wie groß ist die Abweichung vom Erwartungswert?
2. Ausführungsgeschwindigkeit
 - Ist eine Beschleunigung der Anwendung erkennbar?
 - Wie stark sind andere Leistungsindikatoren betroffen? (z.B. Speicherverbrauch)
 - Kann eine Schätzung bezüglich der Skalierbarkeit gegeben werden?
3. Hilfe bei der Parallelisierung
 - Welches Zugriffsmuster ist am besten für eine Parallelisierung geeignet?
 - Kann die Parallelisierung automatisch erfolgen?

Voraussetzungen:

Wir arbeiten mit Visual Studio auf der Basis von C# und dem .NET-Framework.

Die Spracherweiterung soll als Source-To-Source-Erweiterung implementiert werden. Bei Bedarf können Änderungen der .NET-Laufzeitumgebung an der Rotor-Umgebung vorgenommen werden.

Betreuer:

Jochen Schimmel und Korbinian Molitorisz sind wissenschaftliche Mitarbeiter am Lehrstuhl Prof. Walter F. Tichy. Sie befassen sich im Rahmen ihrer Forschung mit den Themen „Parallelisierung sequentieller Softwaresysteme“, „Werkzeugunterstützung zur Identifizierung parallelisierbarer Regionen“, „Werkzeugunterstützung zur automatischen Testfallerzeugung“ und „Erzeugung von Testfallmengen für parallele Anwendungen“.