



14.12.2010

Einladung

Am Donnerstag, dem 6. Januar 2010, Hörsaal M 5

spricht

Prof. Dr. Christoph Kessler (Universität Linköping, Schweden)

10:00 Uhr Lehrvortrag: Sortierverfahren

Zusammenfassung:

Wir lernen drei grundlegende Sortieralgorithmen kennen: Insertion sort (Sortieren durch Einfügen), Selection sort (Sortieren durch Auswahl), und Quicksort. Wir gehen (im Rahmen der zur Verfügung stehenden Zeit) auf die Analyse des Laufzeitbedarfs ein und halten die bei Quicksort verwendeten algorithmischen Entwurfsmuster fest: Divide-and-Conquer und Randomisierung.

10:30 Uhr Forschungsvortrag: Synthese optimierter Programme aus anno-
tierten Komponenten

Zusammenfassung:

Wir beschreiben einen Ansatz zur optimierenden Komposition von Programmen aus annotierten Komponenten. Durch Annotationen erweitern wir die übliche funktionale Schnittstelle von Softwarekomponenten mit zusätzlichen Metadaten und -code, die nichtfunktionale Eigenschaften wie z.B. Laufzeit-, Speicher- oder Energiebedarf der Komponente beschreiben, und mit Mark-up zur expliziten Kennzeichnung der Kompositions- und Variationspunkte im Code. Mit Hilfe der Metadaten können zum Beispiel Laufzeiten durch ein Modell geschätzt oder zur Deployment-Zeit durch Profiling auf der Zielplattform ermittelt werden.

Aus den Metadaten berechnet ein Kompositionswerkzeug zur Deployment-Zeit (off-line) Dispatch- und Scheduling-Tabellen vor und generiert Code, der tabellengesteuert zur Laufzeit den Programmablauf optimiert, indem etwa bei jedem Aufruf einer Komponentenfunktion in Abhängigkeit vom Aufrufkontext (z.B. der aktuellen Problemgrößen und verfügbaren Systemressourcen) die erwartet schnellste Variante unter mehreren äquivalenten Komponenten angesprungen wird, das erwartet schnellste Schedule bei unabhängigen Subtasks gewählt wird, oder Operanden automatisch in die erwartet geeignetste Repräsentation konvertiert werden.

Unser Ansatz stellt eine Generalisierung der zur Zeit sehr populären Autotuning-Verfahren von speziellen Bibliotheksfunktionen auf allgemeine anwenderdefinierte Programmkomponenten dar. Wir demonstrieren anhand mehrerer Fallstudien auf verschiedenen Systemplattformen das Optimierungspotential unserer Methode.

Auf diese Vorträge wird besonders hingewiesen

Matthias Löwe, Dekan