



Static Analysis of Run-time Modes in Kahn Process Networks

vorgelegt von
Diplom-Ingenieur
Michael Beyer

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Peter Pepper
Technische Universität Berlin

Berichtende: Prof. Dr. Sabine Glesner
Technische Universität Berlin

Berichtender: Prof. Dr. Wolf Zimmermann
Martin-Luther-Universität, Halle Wittenberg

Tag der wissenschaftlichen Aussprache: 9. Juli 2012

Berlin 2012

D 83

Berichte aus der Softwaretechnik

Michael Beyer

**Static Analysis of Run-time Modes
in Kahn Process Networks**

D 83 (Diss. TU Berlin)

Shaker Verlag
Aachen 2012

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: Berlin, Techn. Univ., Diss., 2012

Copyright Shaker Verlag 2012

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-1252-1

ISSN 1433-9986

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

In this thesis, we investigate a static analysis of run-time modes in Kahn Process Networks (KPNs) to improve mappings to parallel architectures. In embedded system design, KPNs are used for modeling streaming-oriented applications. To meet the performance and energy constraints of embedded devices, such applications must be mapped to parallel architectures in an optimal way. Finding an optimal mapping to the constrained architecture presumes that the behavior of the KPN is statically known. Considering the run-time behavior of streaming-oriented applications, we can identify run-time modes in which the system offers a certain behavior. In this thesis we present a static analysis that identifies such run-time modes at compile-time.

To that end, we use abstract interpretation based on polyhedral abstraction and an automata-based representation of KPNs to consider the expected run-time behavior. We have developed a state partitioning algorithm to separate control-oriented parts from transformative parts. The automata states are partitioned according to the run-time behavior. We identify modes and their hierarchies for individual processes by using our graph reduction algorithm to transform state partitioned automata to mode automata. To obtain the overall network behavior, we have developed a composition algorithm for asynchronously communicating mode automata. We abstract from unnecessary considerations of interleaving states by choosing one representative execution. The result is a mode automaton describing the whole network behavior.

To evaluate our method, we have implemented a prototype of our analysis. For an arbitrary KPN, our implementation statically analyzes its run-time modes and their corresponding behaviors. We have evaluated typical case studies, e. g., a reconfigurable Finite Impulse Response filter, which demonstrate the practicability of our approach. Optimizing backends that map KPNs to parallel architectures can be guided by these analysis results.

Zusammenfassung

Um die Abbildung von nebenläufigen Anwendungen auf parallele Architekturen zu verbessern, haben wir eine statische Analyse entwickelt, die Laufzeitmodi in Kahn-Prozess-Netzwerken (KPNs) identifiziert. Im Entwurf von eingebetteten Systemen werden KPNs zur Modellierung von *streaming*-orientierten Anwendungen verwendet. Um die Leistungs- und Energiebeschränkungen von eingebetteten Systemen zu erfüllen, müssen diese optimal auf eine parallele Architektur abgebildet werden. Optimale Abbildungen können nur gefunden werden, wenn das Verhalten der Anwendung statisch ist, d.h., nur einen Laufzeitmodus aufweist. In unserem Ansatz betrachten wir die statische Analyse von multimodalen Anwendungen.

Ein gegebenes KPN wird in eine polyhederabstrahierte automatenbasierte Darstellung transformiert. Zur Trennung von kontrollorientierten und funktionalen Anteilen haben wir einen Algorithmus entwickelt, der die Automatenzustände bezüglich des Laufzeitverhaltens partitioniert. Mit Hilfe unseres Graphreduktionsverfahrens identifizieren wir Modi und deren Hierarchien für einzelne Prozesse. Das Gesamtverhalten des Netzwerks bestimmen wir mit unserem Kompositionsalgorithmus für asynchron kommunizierende Automaten. Dabei werden alle Zustandsverschränkungen, die bei nebenläufigen Prozessen auftreten können, durch Repräsentanten abstrahiert. Das Ergebnis ist ein Automat, der das Gesamtverhalten des Netzwerks beschreibt.

Um die praktische Anwendbarkeit unseres Ansatzes zu zeigen, haben wir unsere Analyse implementiert und anhand von typischen Fallbeispielen evaluiert. Optimierende Übersetzer können die Analyseergebnisse nutzen, um KPNs möglichst optimal auf eine parallele Architektur abzubilden.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Fachgebiet “Programmierung eingebetteter Systeme” unter der Leitung von Prof. Dr. Sabine Glesner. Bei ihr möchte ich mich ganz besonders für die Betreuung meiner Arbeit bedanken und dafür, dass sie mich immer unterstützt und mir sehr viel beigebracht hat. Meinem zweiten Gutachter, Prof. Dr. Wolf Zimmermann, bin ich ebenfalls sehr dankbar verbunden für seine freundliche Unterstützung und konstruktiven Anmerkungen.

Meinen Kollegen danke ich für die nette Arbeitsatmosphäre und die anregenden Diskussionen, die das Entstehen der Arbeit vorangetrieben haben. Ganz besonders bedanke ich mich bei Dirk Tetzlaff, meinem Büronachbarn, und Marcel Pockrandt, mit denen ich viele Ideen und Lösungsansätze besprechen konnte. Darüber hinaus bedanke ich mich bei Dr. Paula Herber, Dr. Thomas Göthel und Björn Bartels für das Gegenlesen der Arbeit und für das hilfreiche *Feedback*. An dieser Stellen möchte ich auch nochmal allen Akteuren im Promotionsfilm danken, der ist wirklich toll geworden.

Ein besonderer Dank gilt meiner Familie, all meinen Freunden und meiner Freundin, Sabine Troppens, für den Rückhalt und Beistand. Ein außerordentlicher Dank gilt meinen Eltern, Elisabeth Beyer und Klaus Beyer, für das Erschließen und die Förderung vieler Interessensgebiete.

Michael Beyer, Juli 2012

Contents

1	Introduction	13
1.1	Problem and Objectives	13
1.2	Static Analysis of Run-time Modes	14
1.3	Motivation	15
1.4	Main Contributions	16
1.5	Overview of this Thesis	16
2	Background	19
2.1	Static Program Analysis	19
2.1.1	Data-Flow Analysis	19
2.1.2	Abstract Interpretation	22
2.1.3	Polyhedral Abstraction	24
2.2	Programming Parallel Architectures	28
2.2.1	Parallelism	28
2.2.2	Parallel Architectures	30
2.2.3	Parallel Programming Models	31
2.2.4	Dataflow Networks	32
2.2.5	Kahn Process Networks	34
2.3	Summary	35
3	Related Work	37
3.1	Modeling of Multi-Mode Systems	37
3.2	Simulation and Profiling	39
3.3	Mapping KPNs to Parallel Architectures	40
3.4	Static Program Analysis and Concurrency	41
3.5	Summary	41
4	Static Analysis of Modes	43
4.1	Mapping to Parallel Architectures	43

4.2	Run-time Modes in KPNs	45
4.2.1	Modes in Single Processes	46
4.2.2	Modes in KPNs	48
4.3	Approach	50
5	Polyhedral Abstraction	53
5.1	While Language for KPNs	53
5.2	Control Flow Automata	55
5.3	Polyhedral Abstraction of Paths	57
5.3.1	Communication Signatures	58
5.3.2	Polyhedral Abstraction of Executed Paths	60
5.3.3	Decomposition of Boolean Expressions	63
5.4	Data Augmented Control Flow Automata	65
5.5	Summary	68
6	Mode Analysis for KPNs	69
6.1	State Partitioning for a Single Automaton	69
6.1.1	Introductory Example	70
6.1.2	The State Partitioning Algorithm	71
6.1.3	Separation of Intersecting Preconditions	75
6.2	Identification of Run-time Modes	77
6.2.1	Modes and Mode Automata	77
6.2.2	Mode Identification by Graph Reduction	80
6.3	Composition of Mode Automata	86
6.3.1	Synchronous Composition	87
6.3.2	Asynchronous Composition	95
6.4	Summary	101
7	Evaluation of our Approach	103
7.1	Implementation	103
7.1.1	Polyhedral Abstraction	104
7.1.2	Mode Extraction	106
7.1.3	Composition of Mode Automata	107
7.2	Case Studies	108
7.3	Performance Evaluation	111
7.3.1	FIFO Implementations	111
7.3.2	Scalability	112
7.4	Summary	114

8 Conclusion and Future Work	115
8.1 Results	115
8.2 Discussion	117
8.3 Outlook	118
Bibliography	121
List of Figures	135
List of Algorithms	139