



Universiteit  
Leiden  
The Netherlands

## **Semi-partitioned scheduling and task migration in dataflow networks**

Cannella, E.

### **Citation**

Cannella, E. (2016, October 11). *Semi-partitioned scheduling and task migration in dataflow networks*. Retrieved from <https://hdl.handle.net/1887/43469>

Version: Not Applicable (or Unknown)

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/43469>

**Note:** To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/43469> holds various files of this Leiden University dissertation

**Author:** Cannella, Emanuele

**Title:** Semi-partitioned scheduling and task migration in dataflow networks

**Issue Date:** 2016-10-11

# **Semi-partitioned Scheduling and Task Migration in Dataflow Networks**

Emanuele Cannella



# **Semi-partitioned Scheduling and Task Migration in Dataflow Networks**

## **PROEFSCHRIFT**

ter verkrijging van  
de graad van Doctor aan de Universiteit Leiden,  
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,  
volgens besluit van het College voor Promoties  
te verdedigen op dinsdag 11 oktober 2016  
klokke 10.00 uur

door

Emanuele Cannella  
geboren te Udine, Italië  
in 1983

<b>Promotor:</b>	Prof. dr. Ed F. Deprettere	Universiteit Leiden
<b>Co-Promotor:</b>	Dr. Todor P. Stefanov	Universiteit Leiden
<b>Promotion Committee:</b>	Prof. dr. Peter Marwedel	Technische Universität Dortmund
	Prof. dr. Luigi Raffo	Università di Cagliari
	Dr. Andy Pimentel	Universiteit van Amsterdam
	Prof. dr. Aske Plaat	Universiteit Leiden
	Prof. dr. Harry Wijshoff	Universiteit Leiden
	Prof. dr. Joost Kok	Universiteit Leiden

Semi-partitioned Scheduling and Task Migration in Dataflow Networks  
Emanuele Cannella. -  
Dissertation Universiteit Leiden. - With ref. - With summary in Dutch.

Copyright © 2016 by Emanuele Cannella. All rights reserved.

Cover designed by Shanshan Yang.

This dissertation was typeset using  $\text{\LaTeX}$  and version controlled using Git.

Printed by CPI-Koninklijke Wöhrmann – Zutphen.

*Al nostro angelo Silvana  
e al mitico Franz*





# Contents

<b>Table of Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Trends in Embedded MPSoC Design . . . . .	3
1.1.1 Programming for Multiprocessors: Models of Computation . . .	4
1.1.2 Communication Infrastructures: Networks-on-Chip . . . . .	8
1.2 Challenges in Embedded MPSoC Design . . . . .	10
1.2.1 System Adaptivity . . . . .	10
1.2.2 Timing Requirements . . . . .	12
1.2.3 Cost . . . . .	15
1.2.4 Energy Efficiency . . . . .	16
1.3 Problems Addressed in this Thesis . . . . .	17
1.3.1 Best-Effort Systems . . . . .	17
1.3.2 Hard Real-Time Systems . . . . .	18
1.4 Research Contributions . . . . .	19
1.4.1 Exploiting Task Migration to achieve System Adaptivity in Best-Effort Systems . . . . .	19
1.4.2 Exploiting Semi-partitioned Approaches in Hard Real-Time Scheduling of (C)SDF Graphs . . . . .	21
1.5 Thesis Organization . . . . .	22
<b>2 Background</b>	<b>23</b>
2.1 Dataflow Models of Computation . . . . .	24
2.1.1 (Homogeneous) Synchronous Dataflow ((H)SDF) . . . . .	24
2.1.2 Cyclo-Static Dataflow (CSDF) . . . . .	26
2.1.3 Polyhedral Process Network (PPN) . . . . .	28
2.2 Real-time Scheduling Theory . . . . .	31
2.2.1 Real-time periodic and sporadic task models . . . . .	31
2.2.2 System model . . . . .	31
2.2.3 Multiprocessor Real-Time Scheduling Algorithms . . . . .	32

2.2.4	Uniprocessor Schedulability Analysis . . . . .	33
2.2.5	Multiprocessor Schedulability Analysis . . . . .	34
2.2.6	Partitioning Heuristics . . . . .	36
2.2.7	EDF-fm Semi-partitioned Algorithm . . . . .	37
2.2.8	EDF-os Semi-partitioned Algorithm . . . . .	41
2.3	HRT Scheduling of Acyclic CSDF Graphs [BS11,BS12] . . . . .	42
<b>3</b>	<b>PPN Communication on Networks-on-chip</b>	<b>49</b>
3.1	Problem Statement . . . . .	50
3.2	Contributions . . . . .	52
3.3	Related Work . . . . .	52
3.4	PPN Communication Approaches . . . . .	54
3.4.1	Virtual Connector approach (VC) . . . . .	55
3.4.2	Virtual Connector with Variable Rate approach (VRVC) . . . . .	57
3.4.3	Request-driven approach (R) . . . . .	60
3.5	Case Studies . . . . .	61
3.5.1	Sobel filter . . . . .	62
3.5.2	M-JPEG encoder . . . . .	63
3.5.3	Platform setup . . . . .	63
3.6	Experimental Results . . . . .	65
3.6.1	Inter-tile communication efficiency . . . . .	65
3.6.2	System adaptivity support . . . . .	67
3.7	Discussion . . . . .	69
<b>4</b>	<b>Process Migration Mechanism in a Mapped PPN</b>	<b>71</b>
4.1	Problem Statement . . . . .	71
4.2	Contributions of this Chapter . . . . .	72
4.3	Related Work . . . . .	72
4.4	Proposed Migration Approach . . . . .	74
4.5	Process Migration . . . . .	75
4.5.1	Migratable PPN process structure . . . . .	77
4.5.2	Process migration mechanism . . . . .	81
4.6	Experiments and Results . . . . .	83
4.6.1	Process migration benefits and overhead . . . . .	84
4.7	Discussion . . . . .	86
<b>5</b>	<b>Semi-partitioned Scheduling of CSDF-modeled Streaming Applications</b>	<b>87</b>
5.1	Proposed Extension of the Scheduling Framework of [BS11,BS12] . . . . .	89
5.1.1	Choice of the EDF-fm Semi-partitioned Algorithm . . . . .	89
5.1.2	Implications of Using EDF-fm . . . . .	90
5.2	Problem Statement . . . . .	90
5.3	Contributions . . . . .	91
5.4	Related Work . . . . .	94
5.5	Soft Real-time Scheduling Analysis . . . . .	94
5.5.1	Earliest Start Times in Presence of Tardiness . . . . .	94

---

5.5.2	Minimum Buffer Sizes in Presence of Tardiness . . . . .	97
5.6	FFD-SP Task Assignment Heuristic . . . . .	98
5.7	Evaluation . . . . .	101
5.8	Discussion . . . . .	104
<b>6</b>	<b>Energy Efficient Semi-Partitioned Scheduling of SDF Graphs</b>	<b>105</b>
6.1	Problem Statement . . . . .	106
6.2	Contributions . . . . .	106
6.3	Scope of work . . . . .	108
6.3.1	Assumptions . . . . .	108
6.3.2	Limitations . . . . .	109
6.4	Related work . . . . .	110
6.5	System Model . . . . .	113
6.6	Example of SRT Scheduling of an SDF Graph . . . . .	113
6.7	Proposed Semi-partitioned Algorithm: EDF-ssl . . . . .	114
6.7.1	Assignment Phase . . . . .	117
6.7.2	Execution Phase . . . . .	119
6.7.3	Tardiness Bounds under Fixed Processor Speed . . . . .	120
6.7.4	Tardiness Bounds under PWM Scheme . . . . .	122
6.8	Start times and buffer sizes under EDF-ssl . . . . .	125
6.9	Evaluation . . . . .	128
6.9.1	Power Model . . . . .	128
6.9.2	Energy per Iteration Period . . . . .	129
6.9.3	Experimental Results . . . . .	130
6.10	Discussion . . . . .	134
<b>7</b>	<b>Summary and Discussion</b>	<b>135</b>
7.1	Thesis Summary . . . . .	135
7.2	Discussion . . . . .	137
7.2.1	Assessing the migration mechanism in an industrially- relevant case study . . . . .	137
7.2.2	Application of Chapters 5 and 6 to Daedalus <sup>RT</sup> . . . . .	139
7.2.3	Application models considered in Chapters 3 to 6 . . . . .	142
	<b>Bibliography</b>	<b>145</b>
	<b>List of abbreviations</b>	<b>159</b>
	<b>Samenvatting</b>	<b>161</b>
	<b>List of publications</b>	<b>163</b>
	<b>Curriculum Vitae</b>	<b>165</b>
	<b>Acknowledgments</b>	<b>167</b>



# List of Figures

1.1	System-level design methodology. . . . .	3
1.2	Example of MoC-based application specification (a) and mapping to a platform with 4 PEs (b). . . . .	5
1.3	Comparison of the MoCs considered in this thesis. . . . .	7
1.4	Structure of a NoC composed of 6 tiles with a regular 2D mesh topology. . . . .	9
1.5	Sketch of the hardware/software stack of a computing system. . . . .	11
1.6	Software stack proposed to achieve adaptivity in BE systems. . . . .	20
2.1	Example of an SDF graph composed of actors $A_1, A_2, A_3$ and edges $e_1, e_2$ . . . . .	25
2.2	Example of a CSDF graph. . . . .	27
2.3	Example of a PPN topology and internal structure of process. . . . .	29
2.4	EDF-fm assignment of the task set considered in Example 2.2.1. . . . .	40
2.5	Release pattern of jobs of task $\tau_3$ between processors $\pi_1$ and $\pi_2$ , according to the share assignment of $\tau_3$ in Figure 2.4. . . . .	41
2.6	Hard real-time scheduling of the CSDF graph in Figure 2.2 on page 27 derived using the methodology of [BS11,BS12]. . . . .	45
3.1	Producer-consumer pair which communicates through FIFO buffer $B$ , and corresponding mapping onto a NoC-based platform. . . . .	51
3.2	Producer-consumer pair using the virtual connector method. . . . .	54
3.3	Pseudocode of the VC approach. . . . .	56
3.4	Producer-consumer implementation when using VRVC and R approaches. . . . .	56
3.5	Communication sequence of a producer-consumer pair, using the VRVC approach. . . . .	59
3.6	Pseudocode of the VRVC approach. . . . .	60
3.7	Pseudocode of the Request-driven (R) approach. . . . .	61
3.8	PPN specification of the Sobel filter. . . . .	62
3.9	PPN specification of the M-JPEG encoder. . . . .	63
3.10	Part of the NoC platform structure. . . . .	64
3.11	Structure of middleware- and network- level messages. . . . .	64
3.12	Fixed mappings for Sobel (a) and M-JPEG (b) to test the different PPN communication approaches. . . . .	65

3.13	Total execution time of the M-JPEG and Sobel applications obtained with different MW approaches, and performance comparison with customized systems based on point-to-point connections. . . . .	66
3.14	Traffic injected into the NoC by executing Sobel with different MW approaches. . . . .	67
3.15	Execution time and generated traffic as a function of the process mapping. . . . .	68
4.1	Software stack proposed to achieve adaptivity in BE systems. . . . .	74
4.2	Example of a migration procedure. . . . .	77
4.3	Structure of PPN process $P_2$ , and corresponding basic code structure used to map $P_2$ onto the considered execution platform. . . . .	78
4.4	Structure of migratable process $P_{\text{mig}}$ . . . . .	80
4.5	M-JPEG process scheduling when running on a single tile. . . . .	84
4.6	M-JPEG process scheduling while migrating $P_2$ using the proposed migration mechanism. . . . .	84
5.1	Scheduling framework proposed in [BS11, BS12]. . . . .	88
5.2	Scheduling framework proposed in this chapter. . . . .	91
5.3	Scheduling framework under both HRT and SRT schedulers. . . . .	93
5.4	Example of the conversion of a set of CSDF actors to a real-time periodic task set using the methodology proposed in [BS11, BS12]. . . . .	96
5.5	Worst-case scheduling of source actor $A_i$ when deriving the start time $S_{i \rightarrow j}$ of the destination actor in presence of tardiness. . . . .	97
6.1	Energy-efficient scheduling technique proposed in this chapter. . . . .	107
6.2	Example of the approach proposed in Chapter 5 to schedule an SDF graph with a scheduler that provides SRT guarantees. . . . .	115
6.3	Share assignment considered in Example 6.7.1 . . . . .	116
6.4	Job executions of $\tau_1 = (C_1 = 3, T_1 = 3)$ , as defined in Example 6.7.1, according to the share assignment of Figure 6.3 . . . . .	116
6.5	Share assignments considered in Example 6.7.2. . . . .	119
6.6	PWM scheme execution. . . . .	123
6.7	Supply function $Z(t)$ . . . . .	123
6.8	SDF actors $A_s$ and $A_d$ with dependency over $e_u$ . . . . .	125
6.9	Analysis of the communication between data-dependent actors when both source and destination actors are implemented as migrating tasks. . . . .	126
7.1	PPN model of the H.264 decoder application. . . . .	138
7.2	Abstracted PPN specification of the H.264 decoder application. . . . .	138
7.3	Structure of the execution platform used in our demo and allocation of process replicas to tiles. . . . .	139
7.4	Example of a process migration performed in our demo. . . . .	140
7.5	Overview of the Daedalus <sup>RT</sup> design flow. . . . .	141

# List of Tables

2.1	Summary of mathematical notation. . . . .	23
2.2	Correspondence between dataflow and real-time theory notations resulting from the methodology of [BS11, BS12]. . . . .	47
3.1	Execution times (in clock cycles) of Sobel functions . . . . .	62
3.2	Execution times (in clock cycles) of M-JPEG functions . . . . .	63
4.1	Middleware table example . . . . .	76
5.1	Comparison of different allocation/scheduling approaches. . . . .	102
6.1	Comparison of different share allocation/scheduling approaches. . . . .	131

