

Synchronized Performance Evaluation Methodologies for Communication Systems

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH
Aachen University zur Erlangung des akademischen Grades eines Doktors der
Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker

Elias David Weingärtner

aus Emmendingen

Berichter:

Prof. Dr.-Ing. Klaus Wehrle

Prof. Alejandro Buchmann, PhD.

Tag der mündlichen Prüfung: 19.6.2013

Reports on Communications and Distributed Systems

edited by
Prof. Dr.-Ing. Klaus Wehrle
Communication and Distributed Systems,
RWTH Aachen University

Volume 8

Elias David Weingärtner

Synchronized Performance Evaluation Methodologies for Communication Systems

Shaker Verlag
Aachen 2013

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.: D 82 (Diss. RWTH Aachen University, 2013)

Copyright Shaker Verlag 2013

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 978-3-8440-2227-8

ISSN 2191-0863

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

Contents

1	Introduction	1
1.1	Requirements for Evaluation Techniques	1
1.2	Performance Evaluation Techniques	2
1.2.1	Network Testbeds / Real-World Measurements	2
1.2.2	Network Simulators	3
1.2.3	Analytic Models	3
1.3	Hybrid Evaluation Methodologies	3
1.4	Challenges in Hybrid Evaluation	4
1.4.1	Synchronization and Integration of Timing Domains	4
1.4.2	Interoperability of Evaluation Techniques and Tools	5
1.4.3	Realistic and Contextual Integration for Hybrid Evaluation Frameworks	5
1.5	Research Questions	6
1.6	Contributions	6
1.7	Thesis Organization	8
2	Fundamentals and Background	9
2.1	An Introduction to the Performance Evaluation of Computer Networks	9
2.1.1	Performance Metrics	11
2.1.2	Performance Evaluation Methodologies	13
2.1.3	Hybrid Performance Evaluation Methodologies	25
2.1.4	A BitTorrent Model for Network Emulation	28
2.1.5	The Synchronization Problem of Network Emulation	32
2.2	Virtual Machines	34
2.2.1	Implementation Aspects of System Virtualization	35
2.2.2	The Xen Hypervisor	37

2.3	Hardware Simulation and Full-System Simulators	39
2.3.1	The Simics Full-System Simulator	40
2.3.2	SystemC	41
2.4	Interim Summary	42
3	Synchronized Hybrid Evaluation	43
3.1	Time Synchronization	45
3.1.1	Execution of System Representations: Assumptions	45
3.1.2	Synchronization Requirements	46
3.1.3	The Synchronization Scheme	47
3.1.4	The Synchronization Protocol	51
3.2	System Representation Components	53
3.2.1	Characteristics of System Representations	53
3.2.2	Virtual Machines (VMs)	54
3.2.3	Discrete Event-based Network Simulation	56
3.2.4	Full-System Simulators	58
3.2.5	Other System Representation Types	59
3.2.6	Hybrid System Representations	59
3.3	Packet Exchange between System Representations	60
3.3.1	Non-Synchronized Packet Exchange	60
3.3.2	Integration at the Medium Access Control (MAC) layer	61
3.4	Types of Synchronized Hybrid Evaluation setups	62
3.4.1	Synchronized Network Emulation	62
3.4.2	Synchronized Hardware/Network Co-Simulation	63
3.4.3	Other Types of Compositions	64
3.5	Interim Summary	66
4	Synchronized Emulation Frameworks	67
4.1	SliceTime: A Platform for Scalable and Accurate Network Emulation	68
4.1.1	Conceptual Design	68
4.1.2	Implementation of SliceTime	69
4.1.3	Evaluation of SliceTime	75
4.1.4	Interim Summary	80

4.2	Device Driver-enabled Wireless Network Emulation	80
4.2.1	DDWNE Architecture	81
4.2.2	Implementation of DDWNE for 802.11	85
4.2.3	Evaluation	89
4.2.4	Compatibility with unmodified Wireless Software	92
4.2.5	Interim Summary	94
4.3	Application Studies using SliceTime and DDWNE	94
4.3.1	HTTP Use Case	94
4.3.2	AODV routing daemon study	96
4.3.3	A Hybrid Evaluation Platform for BitTorrent systems	98
4.4	Related Work	104
4.4.1	Link Emulation Engines	105
4.4.2	Virtualization-based Emulation Engines	108
4.4.3	Network Simulator-based Emulation Engines	114
4.4.4	Hardware-based Network Emulation	122
4.5	Interim Conclusion	127
5	A hybrid evaluation framework for networked embedded systems	129
5.1	Motivation	129
5.2	Conceptual Design	130
5.2.1	Synchronization Component	130
5.2.2	Network Simulation	131
5.2.3	Virtual Platform (VP)	131
5.3	Implementation	132
5.3.1	The SyncVP Component	133
5.3.2	Virtual Network Chip	133
5.4	Evaluation	134
5.4.1	Virtual Platform Performance	135
5.4.2	Influence of Synchronization Accuracy	135
5.4.3	Influences of Network Traffic on the VP Bus Load	137
5.4.4	Influence of Payload Processing on Round Trip Time	137
5.5	Related Work	138
5.5.1	Hardware Simulator-based Approaches	139

5.5.2	Approaches based on Co-Simulation	142
5.6	Limitations	144
5.6.1	Non-Optimal Integration of Timing	144
5.6.2	Dependency on Proprietary Software	145
5.7	Interim Summary and Conclusion	145
6	Monitoring and Debugging Communication Software using Virtual Time	147
6.1	Motivation	147
6.2	A Distributed Analysis Framework	148
6.2.1	Challenges and Solutions	148
6.2.2	Architecture	149
6.3	Prototype Implementation	151
6.3.1	Scripting Environment	152
6.3.2	Back-end Server	152
6.4	Functional Tests	153
6.4.1	Time Isolation	153
6.4.2	Monitoring State Changes	154
6.5	Related Work	154
6.5.1	Virtual Machine-based Debugging	154
6.5.2	Distributed Debugging	155
6.5.3	Online System Monitoring	156
6.6	Limitations and Potential Future Work	157
6.7	Interim Summary and Conclusion	158
7	Conclusion	159
7.1	Key Results	160
7.2	Conceptual Limitations	162
7.3	Future Research and Possible Extensions	163
7.4	Final Remarks	166
	Bibliography	167

A	Appendix	185
A.1	SliceTime: Additional Timing Measurements	186
A.2	BitTorrent Client Model Implementation	187
A.2.1	Component Interaction	187
A.3	Setup and Application of DDWNE	188
A.3.1	Setting up DDWNE for ns-3	188
A.3.2	Setup of the DDWNE Driver	188
A.4	List of Abbreviations	189

Abstract

Researchers and developers typically rely on analytical techniques, simulation-based approaches or real-world experiments for the performance evaluation of communication systems. All of these methods have unique strengths. Network simulations are well suitable for conducting virtual experiments of very high scale; the deterministic and thus repeatable execution of simulations also make them an essential method for parameter studies. Performance evaluations carried out with real-world systems allow for investigations with the topmost possible level of realism and detail. Finally, analytical techniques abstract away from the technical environment of a communication system and hence enable the deduction of general evidence.

Unfortunately, all performance evaluation techniques also suffer from a number of individual shortcomings. The analytical formalization of complex communication protocols and their behavior is often very difficult. Network simulations model only the essential functionality of a communication system, making it problematic to apply this technique for analyzing resource usage or system-specific effects. Finally, evaluations carried out with real-world software prototypes are often perturbed by uncontrollable conditions in the environment. In addition, the high amount of hardware and manpower required for real-world performance evaluations of larger scale make such trials often very costly.

Hybrid performance evaluation methodologies are a promising approach for overcoming these issues. The concept of *network emulation* combines the flexibility and scalability of network simulation with the credibility and the level of detail associated with performance evaluations using real-world systems. A second important hybrid methodology is *hardware/network co-simulation*. The idea of this methodology is to integrate different simulation tools to bridge between their individual application domains, for instance network simulation and hardware modeling.

We contribute to the field of hybrid evaluation tools for communication systems in different ways. First, we introduce the concept of *Synchronized Hybrid Evaluation* that generalizes the idea of hybrid performance evaluation and thus is able to subsume a number of existing hybrid techniques. The second central cornerstone of our work is the synchronization and the virtualization of time. Existing network emulation frameworks require real-time capable simulations, which limits the application of network emulation to a rather narrow set of scenarios. Our work on *Synchronized Network Emulation* removes this burden by synchronizing the execution of the network simulation with virtualized communication systems. A further contribution to the field of network emulation is our work on *Device Driver-enabled Wireless Network Emulation*. It tightly integrates the simulated environment with the operating system context of the software prototype. As we further elaborate in this dissertation, both concepts and their respective implementations substantially extend the applicability of network emulation.

In addition, we contribute to the field of hybrid evaluation techniques with two additional frameworks. First, we propose the integration of a SystemC-based hardware simulator with a network simulation tool, aiming at a network centric design of embedded systems. Second, we show that virtualizing the progression of time and synchronizing the execution of virtualized software prototypes is helpful for building distributed debugging and monitoring tools.

Kurzfassung

Die Leistungsbewertung von Kommunikationssystemen wird derzeit meist entweder mit mathematisch-analytischen oder simulativen Methoden durchgeführt; ebenfalls sind Messungen mit realen Prototypen in Textnetzwerken (Test-Beds) verbreitet. All diese Ansätze besitzen individuelle Stärken: Die hohe Skalierbarkeit von Simulationen ermöglicht es, verteilte Kommunikationssysteme oder große Topologien zu untersuchen. Da Netzwerksimulationen eine deterministische Ausführung erlauben, ist es überdies möglich, Simulationsläufe beliebig zu wiederholen, was die Durchführung von Parameterstudien sehr vereinfacht. Im Gegensatz dazu liefern Messungen mit realen Prototypen Ergebnisse hoher Detailtreue und Güte. Mathematisch-Analytische Verfahren können generelle Aussagen über das Systemverhalten unabhängig von der konkreten Technologie liefern.

Ein großes Problem sind jedoch die individuellen Einschränkungen dieser Verfahren. So ist die mathematisch-analytische Formalisierung von komplexen Kommunikationssystemen sehr schwierig und oft nur eingeschränkt machbar. Der Abstraktionsgrad von Simulationen erschwert die Analyse von komplexem Systemverhalten, das sich beispielsweise aus der Interaktion mit dem Betriebssystem ergibt. Während Messungen mit realen Systemen solchen Einschränkungen nicht unterliegen, sind Experimente auf Testbeds typischerweise durch den hohen Hardwarebedarf sehr teuer. Ein großes Problem sind hier ebenfalls eine Vielzahl von schwer zu kontrollierenden externen Einflüssen, beispielsweise Hintergrundverkehr im Netzwerk oder eine nicht-deterministische Mobilität der Netzwerkknoten.

Ein Weg, den individuellen Schwächen dieser Verfahren zu begegnen, sind hybride Ansätze zur Leistungsbewertung. So vereint das Konzept der Netzwerkemulation die Stärken von Simulationen mit der Detailtreue bei Messungen mit realen Systemen. Ein weiterer solcher Ansatz ist das Konzept des Hardware-Software Co-Designs, in dem verschiedene Leistungsbewertungs- und Hardware-Entwurfsmethodiken in einem hybriden Werkzeug zusammengefasst werden.

In dieser Arbeit werden verschiedene Beiträge zu Forschungsfragestellungen im Bereich der hybriden Leistungsbewertungswerkzeuge besprochen. Das Konzept der *synchronisierten hybriden Evaluation* stellt eine Generalisierung hybrider Leistungsbewertungsverfahren dar und erlaubt es, eine Reihe von bestehenden hybriden Messverfahren zu subsumieren. Das Hauptaugenmerk der Arbeit liegt auf einer daraus abgeleiteten Technologie, der *synchronisierten Netzwerkemulation*. Dieses Verfahren erlaubt es, Netzwerkemulationen mit Simulationen beliebiger Laufzeitkomplexität zu kombinieren und durchzuführen. Die *Gerätetreiber-basierte Netzwerkemulation für drahtlose Übertragungsverfahren (DDWNE)* ermöglicht es, den simulativen Kontext eng an die Ausführungsumgebung der realen Prototypen in einem Emulationsszenario zu binden.

In weiteren Verlauf werden zwei neue Evaluations-Werkzeuge vorgestellt. Zunächst wird ein Ansatz für die Netzwerk-zentrierte Entwicklung von eingebetteten Systemen diskutiert, das intern eine Hardware-Simulation auf Basis von SystemC mit einem Ereignis-basierten Simulator kombiniert. Abschließend wird ein Ansatz beschrieben, der mit Hilfe von Basistechnologien aus der synchronisierten Netzwerksimulation eine Umgebung für die verteilte Fehlersuche und Beobachtung der Ausführung von Kommunikationssystemen erlaubt.

Acknowledgments

Foremost I'd like to express my strong gratitude to my advisor, Klaus Wehrle. Six years ago, Klaus supported me in obtaining a PhD scholarship that enabled my journey with the fabulous COMSYS crew. He pointed me to a research problem that evolved into this doctoral thesis. Beyond that, I thank Klaus very much for creating an open-minded, creative and enjoyable environment. I will never forget the barbecues, skiing in Kleinwalsertal and all our intense conversations!

I'd also like to greatly thank Prof. Alejandro Buchmann not only for co-advising this dissertation, but also for all the exciting discussions we had during all the MAKI and QuaP2P-2 meetings in Darmstadt, Aachen and Kleinwalsertal.

This dissertation would not have been possible without the outstanding contributions by the highly talented and gifted students and colleagues I was lucky to work with. I'd like to give exceptional kudos to Florian Schmidt and Hendrik vom Lehn for their countless efforts with SliceTime. I deeply acknowledge the work of Marko Ritter for taming gdbserver-xen and Apache Thrift. René Glebke, Martin Lang and Alexander Hocks deserve a lot of praise for their patience with VODSim. I'd also like to thank Suraj Prabhakaran and Martin Lindner for additional SliceTime measurements. Finally, I'm very grateful for the fruitful cooperation with Stefan Schürmans, who introduced me to the concept of Hardware-Software Co-Design.

I owe the entire COMSYS team a big debt of gratitude. Tobin, I thank you so much for all the scientific and non-scientific discussions and for your exactness and diligence, not only when we've been writing papers together! Jó, I envy your endless curiosity and your enthusiasm! Raimondas and Georg, you have been both wonderful office mates. I have learned a lot from you! Ismet, thank you for being my No. 1 lifesaver by supplying me with coffee and snacks and for many great down-to-earth advices. I appreciate Petra, Ulrike and Rainer for being so helpful with all organizational issues I ran into - and Dirk and Janosch for taking over a lot of common duties in a self-forgetting manner! All of you have always been a wonderful source of inspiration and good reason to procrastinate. I will never forget three of you spontaneously performing "Bochum" by Herbert Grönemeyer in my office!

I am very thankful for the great support of my amazing friends. You have always been there to listen. You have always been great motivators, for example when we traveled the world together or simply by giving me a smile.

Finally, I'd like to express the strongest gratitude to my parents and my entire family. Rainer, Gertrud and Klaus Weingärtner have continuously supported my curiosity from an early age. I also deeply thank my brother Simon for letting me spend so much time on our C64 and all other family computers. This eventually made me become a computer scientist!