COM SYS

Chair of
Communication and Distributed Systems

RWTH AACHEN UNIVERSITY

Mirko Stoffers

Automated Optimization of Discrete Event Simulations without Knowing the Model

Reports on Communications and Distributed Systems
Editor: Prof. Dr.-Ing. Klaus Wehrle

VOL 19

Automated Optimization of Discrete Event Simulations without Knowing the Model

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

Master of Science

Mirko Stoffers

aus Bochum

Berichter:

Prof. Dr.-Ing. Klaus Wehrle
Prof. Richard Fujimoto, Ph.D.

Tag der mündlichen Prüfung: 05. 07. 2019

Reports on Communications and Distributed Systems

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

Volume 19

Mirko Stoffers

Automated Optimization of Discrete Event Simulations without Knowing the Model

Shaker Verlag
Düren 2019

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, 2019)

ISBN 978-3-8440-6973-0
ISSN 2191-0863

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9
Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

Modeling and simulation is an essential element in the research and development of new concepts and products in any domain. The demand for the development of more and more complex systems drives the complexity of the simulation models as well. If such simulations are not executed efficiently, overly long execution times hamper conduction of necessary experiments. We identified two major types of resources that must be used efficiently to reduce simulation runtime. On the one hand, multiple computing instances (e. g., CPU cores) can be used to distribute the workload and perform independent computations simultaneously. On the other hand, workload stemming from redundant computations can be avoided altogether by exploiting the presence of unused main memory to store intermediate results.

We observe that typically in the development cycle of products and simulation models neither time and resources nor required expertise is available to apply sophisticated runtime optimization manually. We conclude that it is of utmost importance to investigate approaches to speed up simulation automatically. The most prevalent challenge of automating optimization is that, at the time of researching and developing the acceleration concepts and tools, the model is not yet available, and we need to assume that the model will not be implemented for a specific optimization technique. Hence, our methodologies need to be devised without the model at hand, which can only be used by the finally implemented optimization tool at its runtime. In this thesis, we investigate how computer simulations can be automatically accelerated using each of the two optimization potentials mentioned above (multiple computing instances or available memory) without the model being provided at the time of researching the concepts and developing the tools.

For the utilization of multiple computing instances we devise methodologies to automatically derive data-dependencies directly from the model implementation itself, enabling to discover more independent, hence parallelizable, work items. We investigate the capabilities of doing so either statically—while compiling the model implementation—or dynamically—while the simulation runs. The gathered knowledge can be used not only in conservatively synchronized parallel simulations but also enables to dynamically switch to more optimistic paradigms.

For the utilization of available memory we explore the opportunity to avoid redundant computations altogether. We base this on the observation that such redundancies occur frequently in many simulations and simulation parameter studies. Our approach to automatically avoid redundant computations operates on almost arbitrary input code, hence being generally applicable, especially in the modeling and simulation domain.

By combining the two optimization vectors we unleash the full power of both at the same time. We demonstrate that our approaches are able to accelerate the simulations in a parameter study by a factor of more than 600 such that the entire study—including fixed setup and teardown efforts—can execute more than 200 times faster than without optimization. We conclude that the methodologies discussed in this thesis demonstrate the potential to speed up simulations without prior knowledge about the model to optimize.

Kurzfassung

Die Durchführung von Simulationen ist ein zentraler Bestandteil der Erforschung und Entwicklung neuer Konzepte und Produkte in vielen Bereichen. Der Bedarf an komplexeren Systemen erhöht zwangsläufig die Komplexität der Simulationsmodelle. Zu lange Laufzeiten durch ineffiziente Ausführung verhindern dann die Durchführung wichtiger Experimente. Dies betrifft im Wesentlichen zwei Arten von Ressourcen, die effizient genutzt werden müssen um die Laufzeit zu reduzieren: Einerseits können mehrere Recheneinheiten (z. B. CPU-Kerne) verwendet werden, um unabhängige Berechnungen zu verteilen und gleichzeitig durchzuführen. Andererseits können Berechnungen vollständig vermieden werden, wenn diese redundant sind und ungenutzter Speicher zur Ablage von Zwischenergebnissen genutzt werden kann.

Da im Entwicklungszyklus der Simulationsmodelle oft weder Zeit und Ressourcen noch das notwendige Expertenwissen zur Verfügung stehen, die Laufzeit manuell zu optimieren, sind Ansätze zur automatischen Beschleunigung der Simulation unabdingbar. Jedoch steht weder zum Zeitpunkt der Erforschung und Entwicklung der Beschleunigungskonzepte und -werkzeuge das Modell bereits zur Verfügung noch kann davon ausgegangen werden, dass das Modell auf die entwickelte Optimierungstechnik hin implementiert werden wird. Die Methoden müssen also ohne vorliegendes Modell entwickelt werden, welches dann erst zur Laufzeit des bereits implementierten Optimierungswerkzeugs zur Verfügung stehen wird. Diese Arbeit untersucht die Frage, wie Computersimulationen entlang der beiden oben genannten Optimierungsvektoren (mehrere Recheneinheiten oder ungenutzter Speicher) automatisch beschleunigt werden können, ohne dass das Modell bereits zum Zeitpunkt der Erforschung der Konzepte und der Entwicklung der Werkzeuge bereitstehen muss.

Zur Nutzung mehrerer Recheneinheiten werden Methodiken untersucht, Datenabhängigkeiten automatisch aus dem Modell selbst abzuleiten, um eine größere Zahl an unabhängigen, also parallelisierbaren Aufgaben zu identifizieren. Dies geschieht sowohl mittels statischer Analyse beim Kompilieren des Modells als auch mittels dynamischer Methoden während der Simulation. Die ermittelten Informationen können nicht nur in konservativ synchronisierten Simulationen genutzt werden, sondern ermöglichen auch den dynamischen Wechsel zu optimistischen Ansätzen.

Zur Nutzung des ungenutzten Arbeitsspeichers wird untersucht, wie redundante Berechnungen vollständig vermieden werden können. Dies basiert auf der Beobachtung, dass derartige Redundanzen in vielen Simulationen und Parameterstudien häufig vorkommen. Der entwickelte Ansatz zur automatisierten Vermeidung redundanter Berechnungen kann dann auf nahezu beliebigem Programmcode angewandt werden, insbesondere natürlich im Bereich der Simulation.

Schließlich entfaltet die Kombination beider Optimierungsvektoren deren volle Leistungsfähigkeit auf einmal. Es zeigt sich, dass die Simulationen einer Parameterstudie um den Faktor 600 beschleunigt werden können, wodurch die gesamte Studie (einschließlich Auf- und Abbau der einzelnen Simulationen) mehr als 200 mal schneller durchgeführt werden kann als ohne Optimierung. Insgesamt weisen die in dieser Arbeit diskutierten Methoden somit ein erhebliches Potenzial zur Beschleunigung von Simulationen ohne Vorkenntnisse über das zu optimierende Modell auf.

Acknowledgments

In every thing give thanks: for this is the will of God in Christ Jesus concerning you.

—1 Thes. 5:18

This thesis would not have been possible without a lot of valuable contributions of so many people over the course of the last couple of years. I would like to use this opportunity to express my deepest gratitude to everybody who enabled me to create this dissertation.

First of all I want to express my great appreciation to my first advisor Klaus Wehrle for providing the foundations to pursue a PhD in his research group. He provided me the freedom to choose research topics according to my interests, we had many valuable discussions, and I had excellent working conditions at the chair. I am much obliged to James Gross who initially hired me before he moved to Stockholm. We had valuable discussions and motivating conversations not only before he left but also during his visits at Aachen. I am deeply indebted to Richard Fujimoto for agreeing to be my second advisor. Despite being an honored and busy professor he concerned himself with my thesis and traveled all the way over the Atlantic Ocean just for my PhD defense. Additionally, my gratitude goes to Thomas Noll and Bastian Leibe for agreeing to serve on my examination committee and spending their valuable time.

This thesis is compiled of research advances resulting from several collaborations with students and colleagues. I am convinced that it would not exist in its current form without those contributions. My special gratitude goes to all the students who provided the foundations for the papers and contributions the thesis bases upon. Besides that, I had the honor to work with a lot of students whose research efforts eventually did not become part of this dissertation, but all of these collaborations provided me valuable insights and constitute important steps of the entire process. Thank you, to everybody who worked with me during the last couple of years!

A great thank also goes to everybody on the COMSYS team! You have made this chair a place I have always been enjoying working at. Thanks to each of you there is such a nice working atmosphere at the chair. Everybody is always willing to help, be it in scientific discussions, by proof-reading, or whatever I needed. COMSYS is not only a place to work at but also a place to live. Thank you very much to everybody for nice billiard games, chats, celebrations, and all the fun we've been having!

Finally, I would like to thank everybody else who supported me in any way throughout the last couple of years and thus enabled me to eventually conclude my PhD!

Contents