

SpringerBriefs in Computer Science

Series editors

Stan Zdonik, Brown University, Providence, USA
Shashi Shekhar, University of Minnesota, Minneapolis, USA
Jonathan Katz, University of Maryland, College Park, USA
Xindong Wu, University of Vermont, Burlington, USA
Lakhmi C. Jain, University of South Australia, Adelaide, Australia
David Padua, University of Illinois Urbana-Champaign, Urbana, USA
Xuemin (Sherman) Shen, University of Waterloo, Waterloo, Canada
Borko Furht, Florida Atlantic University, Boca Raton, USA
V.S. Subrahmanian, University of Maryland, College Park, USA
Martial Hebert, Carnegie Mellon University, Pittsburgh, USA
Katsushi Ikeuchi, University of Tokyo, Tokyo, Japan
Bruno Siciliano, Università di Napoli Federico II, Napoli, Italy
Sushil Jajodia, George Mason University, Fairfax, USA
Newton Lee, Newton Lee Laboratories, LLC, Tujunga, USA

More information about this series at <http://www.springer.com/series/10028>

Alexander Heinecke · Wolfgang Eckhardt
Martin Horsch · Hans-Joachim Bungartz

Supercomputing for Molecular Dynamics Simulations

Handling Multi-Trillion Particles
in Nanofluidics

 Springer

Alexander Heinecke
Intel Corporation
Santa Clara, CA
USA

Martin Horsch
University of Kaiserslautern
Kaiserslautern
Germany

Wolfgang Eckhardt
Technische Universität München
Garching
Germany

Hans-Joachim Bungartz
Technische Universität München
Garching
Germany

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-17147-0 ISBN 978-3-319-17148-7 (eBook)
DOI 10.1007/978-3-319-17148-7

Library of Congress Control Number: 2015935406

Springer Cham Heidelberg New York Dordrecht London

© The Author(s) 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

Overview and Goals

Molecular dynamics simulation (MD) as such has long matured and has become a valuable tool in a variety of disciplines. For several reasons, however, MD is just about to evolve as a general tool in engineering sciences. In process engineering in particular, the development of MD simulation has become a focus of research during the last years and still gains momentum, slowly undergoing the transition from a scholarly academic occupation to a key technology in industrial-scale fluid process engineering. Central to ongoing efforts in Germany is the development of the simulation code `ls1 mardyn`, which has now been developed for over a decade in an interdisciplinary cooperation of researchers in process engineering high-performance computing (HPC).

It is the purpose of this book to present modern implementations of relevant MD algorithms at the example of `ls1 mardyn`, a simulation code targeting engineering applications. Thereby, we strictly focus on HPC-related aspects. We cover the implementation on HPC architectures, taking Intel Xeon and Intel Xeon Phi clusters as representatives of current platforms. We describe the distributed- and shared-memory parallelization, including load balancing, on these platforms and particularly focus on the efficient implementation of the compute kernels. Thereby, we also describe the software-architecture of the resulting code.

The developments presented here approximately cover work of the years 2008–2014, and have been spread over (at least) three dissertations and several publications. Motivated by the positive perception of our work in the community and the recognition of our work at the International Supercomputing Conference 2013 (ISC'13) with the PRACE ISC Award, we aim to give a comprehensive and unified description of our efforts.

Organization of the Book

In this work, we place the emphasis on the efficient implementation and parallelization of a MD code. In order to build a solid basis to understand the important characteristics of such a simulation code, we cover modeling approaches as far as they are relevant to process engineering, and give a concise description of the basic MD algorithms. Based on that description, we motivate the development of a simulation code specialized on process engineering applications, and discuss the software structure of the code `ls1 mardyn`. To understand the requirements on an efficient implementation of the relevant algorithms on current hardware, we give a rather deep explanation of the Intel Xeon and Intel Xeon Phi. Subsequently, we detail the parallel implementation of MD simulation on these systems making use of shared- and distributed-memory parallelization, including an efficient load-balancing scheme. Most focus is put on the efficient vectorization of the compute kernels for the target processors, using intrinsics. The described implementations are extensively benchmarked through runtime experiments. In addition to the optimized production version of the simulation code `ls1 mardyn`, a hybrid parallel version simultaneously making use of the Intel Xeon Phi co-processor and the Intel Xeon-based host system is evaluated, as well as a version specialized on atomic fluids, which is executed on up to 146,016 cores on the SuperMUC cluster.

Target Audience

This contribution mainly targets readers at the postgraduate and Ph.D. level in the broad field of scientific computing—where MD simulation, process engineering, computer science, and HPC coalesce. We believe that the presented research can serve both as a good starting point for further research as well as a source of ideas about alternative approaches.

Acknowledgments

Finally, we would like to acknowledge the many helpful comments of and discussions with colleagues, which helped us to develop our ideas in the way they did. We also thank all the students who contributed to this work through research projects and theses. Last but not least, we express our gratitude to our colleagues at Springer for accompanying us on the way of writing this book.

Intel Legal Disclaimer

Optimization Notice: Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <http://www.intel.com/performance>. Intel, Xeon, and Intel Xeon Phi are trademarks of Intel Corporation in the U.S. and/or other countries.

Contents

1 Introduction	1
1.1 The Art of Molecular Modeling and Simulation	3
1.2 Focus and Structure of This Work	6
References	8
2 Molecular Dynamics Simulation	11
2.1 Molecular Models and Potentials	11
2.2 Statistical Ensembles	17
2.3 Algorithms for Nearest-Neighbor Search	18
2.4 Characteristics of Large-Scale Molecular Dynamics Simulation of Fluids	21
2.5 Simulation Code Mardyn	23
References	27
3 Parallelization of MD Algorithms and Load Balancing	31
3.1 Target Systems	32
3.2 Shared-Memory Parallelization	37
3.3 Spatial Domain Decomposition	40
3.4 Load Balancing Based on KD-Trees	41
References	43
4 Efficient Implementation of the Force Calculation in MD Simulations	45
4.1 Memory Access Optimizations	46
4.2 Vectorization	48
4.3 Optimization Possibilities for Monatomic Fluids	54
References	57

5 Experiments	59
5.1 Performance on SuperMUC.	60
5.2 Performance on the Intel Xeon Phi Coprocessor.	67
5.3 Multi-Trillion Particle Simulations	70
5.4 Summary	72
References	73
6 Conclusion	75