

SpringerBriefs in Computer Science

Series editors

Stan Zdonik, Brown University, Providence, RI, USA

Shashi Shekhar, University of Minnesota, Minneapolis, MN, USA

Xindong Wu, University of Vermont, Burlington, VT, USA

Lakhmi C. Jain, University of South Australia, Adelaide, SA, Australia

David Padua, University of Illinois Urbana-Champaign, Urbana, IL, USA

Xuemin Sherman Shen, University of Waterloo, Waterloo, ON, Canada

Borko Furht, Florida Atlantic University, Boca Raton, FL, USA

V. S. Subrahmanian, Department of Computer Science, University of Maryland,
College Park, MD, USA

Martial Hebert, Carnegie Mellon University, Pittsburgh, PA, USA

Katsushi Ikeuchi, Meguro-ku, University of Tokyo, Tokyo, Japan

Bruno Siciliano, Dipartimento di Ingegneria Elettrica e delle Tecnologie
dell'Informazione, Università di Napoli Federico II, Napoli, Italy

Sushil Jajodia, George Mason University, Fairfax, VA, USA

Newton Lee, Institute for Education Research and Scholarships, Los Angeles,
CA, USA

SpringerBriefs present concise summaries of cutting-edge research and practical applications across a wide spectrum of fields. Featuring compact volumes of 50 to 125 pages, the series covers a range of content from professional to academic.

Typical topics might include:

- A timely report of state-of-the art analytical techniques
- A bridge between new research results, as published in journal articles, and a contextual literature review
- A snapshot of a hot or emerging topic
- An in-depth case study or clinical example
- A presentation of core concepts that students must understand in order to make independent contributions

Briefs allow authors to present their ideas and readers to absorb them with minimal time investment. Briefs will be published as part of Springer's eBook collection, with millions of users worldwide. In addition, Briefs will be available for individual print and electronic purchase. Briefs are characterized by fast, global electronic dissemination, standard publishing contracts, easy-to-use manuscript preparation and formatting guidelines, and expedited production schedules. We aim for publication 8–12 weeks after acceptance. Both solicited and unsolicited manuscripts are considered for publication in this series.

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

Arthur Francisco Lorenzon
Antonio Carlos Schneider Beck Filho

Parallel Computing Hits the Power Wall

Principles, Challenges, and a Survey
of Solutions

 Springer

Arthur Francisco Lorenzon
Department of Computer Science
Federal University of Pampa (UNIPAMPA)
Alegrete, Rio Grande do Sul, Brazil

Antonio Carlos Schneider Beck Filho
Institute of Informatics, Campus do Vale
Federal University of Rio Grande
do Sul (UFRGS)
Porto Alegre, Rio Grande do Sul, Brazil

ISSN 2191-5768

ISSN 2191-5776 (electronic)

SpringerBriefs in Computer Science

ISBN 978-3-030-28718-4

ISBN 978-3-030-28719-1 (eBook)

<https://doi.org/10.1007/978-3-030-28719-1>

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2019

This work is subject to copyright. All rights are solely and exclusively licensed 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. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*This book is dedicated to the memory of
Márcia Cristina and Aurora Cera.*

Preface

Efficiently exploiting thread-level parallelism from modern multicore systems has been challenging for software developers. While blindly increasing the number of threads may lead to performance gains, it can also result in a disproportionate increase in energy consumption. In the same way, optimization techniques for reducing energy consumption, such as DVFS and power gating, can lead to huge performance loss if used incorrectly. In this book, we present and discuss several techniques that address these challenges. We start by providing a brief theoretical background on parallel computing in software and the sources of power consumption. Then, we show how different parallel programming interfaces and communication models may affect energy consumption in different ways. Next, we discuss tuning techniques to adapt the number of threads/operating frequency to achieve the best compromise between performance and energy. We finish this book with a detailed analysis of a representative example of an adaptive approach.

Alegrete, Brazil
Porto Alegre, Brazil

Arthur Francisco Lorenzon
Antonio Carlos Schneider Beck Filho

Acknowledgments

The authors would like to thank the friends and colleagues at Informatics Institute of the Federal University of Rio Grande do Sul and give a special thanks to all the people in the Embedded Systems Laboratory, who have contributed to this research since 2013.

The authors would also like to thank the Brazilian research support agencies, FAPERGS, CAPES, and CNPq.

Contents

1	Runtime Adaptability: The Key for Improving Parallel Applications ..	1
1.1	Introduction	1
1.2	Scalability Analysis	3
1.2.1	Variables Involved	5
1.3	This Book	7
2	Fundamental Concepts	9
2.1	Parallel Computing in Software	9
2.1.1	Communication Models	9
2.1.2	Parallel Programming Interfaces	10
2.1.3	Multicore Architectures	12
2.2	Power and Energy Consumption	13
2.2.1	Dynamic Voltage and Frequency Scaling	14
2.2.2	Power Gating	15
3	The Impact of Parallel Programming Interfaces on Energy	17
3.1	Methodology	17
3.1.1	Benchmarks	17
3.1.2	Multicore Architectures	19
3.1.3	Execution Environment	20
3.1.4	Setup	22
3.2	Results	23
3.2.1	Performance and Energy Consumption	23
3.2.2	Energy-Delay Product	30
3.2.3	The Influence of the Static Power Consumption	34
3.3	Discussion	38
4	Tuning Parallel Applications	41
4.1	Design Space Exploration of Optimization Techniques	41
4.2	Dynamic Concurrency Throttling	42
4.2.1	Approaches with no Runtime Adaptation and no Transparency	43
4.2.2	Approaches with Runtime Adaptation and/or Transparency ..	45

- 4.3 Dynamic Voltage and Frequency Scaling 49
 - 4.3.1 Approaches with no Runtime Adaptation and no Transparency 49
 - 4.3.2 Approaches with Runtime Adaptation and/or Transparency .. 50
- 4.4 DCT and DVFS 51
 - 4.4.1 Approaches with no Runtime Adaptation and no Transparency 51
 - 4.4.2 Approaches with Runtime Adaptation and/or Transparency .. 52
- 5 Case Study: DCT with Aurora 55**
 - 5.1 The Need for Adaptability and Transparency 55
 - 5.2 Aurora: Seamless Optimization of OpenMP Applications..... 56
 - 5.2.1 Integration to OpenMP 56
 - 5.2.2 Search Algorithm 60
 - 5.3 Evaluation of Aurora..... 63
 - 5.3.1 Methodology 63
 - 5.3.2 Results..... 66
- 6 Conclusions 79**
- References..... 81**

Acronyms

CMOS	Complementary metal oxide semiconductor
DCT	Dynamic concurrency throttling
DSE	Design space exploration
DVFS	Dynamic voltage and frequency scaling
EDP	Energy-delay product
FFT	Fast fourier transform
FIFO	First-in first-out
FU	Function unit
GPP	General-purpose processors
HC	High communication
HPC	High-performance computing
ILP	Instruction-level parallelism
ISA	Instruction set architecture
LC	Low communication
MPI	Message passing interface
OpenMP	Open multi-programming
PAPI	Performance application programming interface
PPI	Parallel programming interface
PThreads	POSIX threads
RAM	Random access memory
SMT	Simultaneous multithreading
SoC	System-on-chip
TBB	Threading building blocks
TDP	Thermal design power
TLP	Thread-level parallelism