Lecture Notes in Computer Science 8913

Rachid Guerraoui   Paolo Romano (Eds.)

Transactional Memory

Foundations, Algorithms, Tools, and Applications

COST Action Euro-TM IC1001

Springer

Volume Editors

Rachid Guerraoui
EPFL, Lausanne, Switzerland
E-mail: rachid.guerraoui@epfl.ch

Paolo Romano
INESC-ID, Lisboa, Portugal
E-mail: romano@inesc-id.pt

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-319-14719-2 e-ISBN 978-3-319-14720-8
DOI 10.1007/978-3-319-14720-8
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014958340

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Acknowledgments: Euro-TM is an Action supported by the COST Association that has gathered researchers from 17 European countries and over 40 institutions. All the authors of the book have actively participated in the activities of Euro-TM.
The editors are grateful to the COST programme for supporting the Euro-TM initiative (IC1001 - Transactional Memories: Foundations, Algorithms, Tools, and Applications), a forum that was fundamental for the preparation of this book. The editors also express their gratitude to all authors, for their enthusiastic and meticulous cooperation in the preparation of this book. A special thank goes to Maria Couceiro for her valuable support with the editing of the book.
This publication is supported by COST.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

COST Description

COST - European Cooperation in Science and Technology is an intergovernmental framework aimed at facilitating the collaboration and networking of scientists and researchers at European level. It was established in 1971 by 19 member countries and currently includes 35 member countries across Europe, and Israel as a cooperating state.

COST funds pan-European, bottom-up networks of scientists and researchers across all science and technology fields. These networks, called 'COST Actions', promote international coordination of nationally-funded research.

By fostering the networking of researchers at an international level, COST enables break-through scientific developments leading to new concepts and products, thereby contributing to strengthening Europe's research and innovation capacities.

COST's mission focuses in particular on:

- Building capacity by connecting high quality scientific communities throughout Europe and worldwide;
- Providing networking opportunities for early career investigators;
- Increasing the impact of research on policy makers, regulatory bodies and national decision makers as well as the private sector.

Through its inclusiveness policy, COST supports the integration of research communities in less research-intensive countries across Europe, leverages national research investments and addresses societal issues.

Over 45 000 European scientists benefit from their involvement in COST Actions on a yearly basis. This allows the pooling of national research funding and helps countries research communities achieve common goals.

As a precursor of advanced multidisciplinary research, COST anticipates and complements the activities of EU Framework Programmes, constituting a "bridge" towards the scientific communities of emerging countries.

Traditionally, COST draws its budget for networking activities from successive EU RTD Framework Programmes.

COST Association Legal Notice

COST is supported by the EU Framework Programme Horizon 2020

To waves and wind,
to the marvel and the fury of elements,
to the search for perfection.

Preface

Parallel programming (PP) used to be an area once confined to a few niches, such as scientific and high-performance computing applications. However, with the proliferation of multicore processors, and the emergence of new, inherently parallel and distributed deployment platforms, such as those provided by cloud computing, parallel programming has definitely become a mainstream concern.

Unfortunately, writing scalable parallel programs using traditional lock-based synchronization primitives is well known to be a hard, time-consuming, and error-prone task, mastered by a minority of programmers only. Thus, to bring parallel programming into the mainstream of software development, we are in urgent need of better programming models.

Building on the abstraction of atomic transactions, and freeing programmers from the complexity of conventional synchronization schemes, transactional memories (TMs) promise to respond exactly to this need, simplifying the development and verification of concurrent programs, enhancing code reliability, and boosting productivity.

Over the last decade, TM has been subject to intense research in computer science and engineering, with hundreds of papers published in prestigious international conferences and journals addressing a wide range of complementary aspects including hardware and operating systems (OSs) support, language integration, as well as algorithms and theoretical foundations. Moreover, even if TMs were first introduced in the context of shared-memory multiprocessors, the abstraction of distributed transactional memory (DTM) has been garnering growing interest of late in the area of cloud computing, due to two main reasons. On the one hand, the increasing reliance of cloud computing platforms on in-memory data management techniques, fostered by the need for maximizing efficiency and supporting real-time data processing of ever-growing volumes of data. On the other hand, the inherent complexity of building large-scale applications on top of the weakly consistent models embraced by the first generation of large-scale NoSQL data stores, which has motivated intensive research on mechanisms aimed to enforce strong consistency semantics in large-scale distributed platforms.

The growing interest in TMs has not only been confined to academic environments. On the industrial side, some of the major players of the software and hardware markets have been up-front in the research and development of prototypal products providing support for TMs. The maturing of research in the field of TM has recently led to hardware TM implementations on several mainstream commercial microprocessors and to the integration of TM support for the world's leading open source compiler.

In such a vast inter-disciplinary domain, the Euro-TM COST Action (IC1001) has served as a catalyzer and a bridge for the various research communities looking at disparate, yet subtly interconnected, aspects of TM. Starting in February 2011 and ending in February 2015, Euro-TM established a Pan-European research network that brought together more than 250 researchers from 17 European countries. Over the last 4 years Euro-TM helped shape the research agenda in the TM area and ensured timely dissemination of research results via thematic workshops; fostered joint publications and research projects; showcased relevant research results in the area in industrial conferences; and educated early-stage researchers via doctoral schools hosting renowned specialists both from industry and academy.

This book emerged from the idea to have Euro-TM experts compile recent results in the TM area in a single and consistent volume. Contributions have been carefully selected and revised to provide a broad coverage of several fundamental issues associated with the design and implementation of TM systems, including their theoretical underpinnings and algorithmic foundations, programming language integration and verification tools, hardware supports, distributed TMs, self-tuning mechanisms, as well as lessons learnt from building complex TM-based applications.

Organization. The book is organized in four sections. The four chapters encompassed by Section 1 address theoretical foundations of TM systems, focusing not only on the definition of safety (Chaps. 1 and 3) and liveness (Chap. 2) guarantees in TM, but also on the problem of designing disjoint-access (i.e., inherently scalable) TM implementations (Chap. 4).

Section 2 covers algorithmic aspects associated with the design of TM systems. First, it illustrates the main algorithmic alternatives in the design space of software (Chap. 5) and hardware (Chap. 6) implementations of TM. Next, it addresses the design and implementation of multi-version TM (Chaps. 7 and 8). Finally, it surveys existing techniques in the literature on TM support for parallel nesting (Chap. 9).

Section 3 deals with the issue of mitigating or even avoiding contention in TM systems, by leveraging on various policies for contention management and transaction scheduling (Chap. 10), as well as techniques that attempt to leverage on semantic knowledge on the code to be executed transactionally (Chap. 11).

Section 4 investigates aspects related to TM and reliability from a twofold perspective: how to leverage on the TM abstraction to simplify exception handling in programming languages (Chap. 12) and mask exceptions at the hardware level (Chap. 13); how to ensure the correctness of TM-based applications by means of automatic verification tools (Chap. 14).

Section 5 addresses the design of DTM platforms. Chapter 15 first provides background on algorithms for replication of transactional systems, by focusing on fully replicated systems (in which each node maintains a full copy of the system state). Chapter 16 considers partially replicated TM systems, analyzing trade-offs between the data-flow and control-flow paradigms. The section is concluded by a survey of existing protocols for DTM systems that adopt the data-flow paradigm.

Finally, Section 6 covers the development of TM-based applications, and on their self-tuning. In more detail, Chap. 17 reports the lessons learnt while developing complex TM-based applications. Chapters 18 and 19 focus instead on self-tuning techniques for TM systems. The former addresses the issue of dynamically adapting the degree of parallelism in shared-memory TM systems. The latter discusses the key challenges associated with the design of self-tuning mechanisms in DTM platforms, surveying existing work in this area.

In summary, the present book provides a unique collection of tutorials, which will allow graduate students in computer science to get familiar with state-of-the-art research in TM and its applications, typically along their road to doctoral studies.

It is noteworthy that this book has only been made possible thanks to the dedication and expertise of our contributing authors, many of whom are from the COST Action Euro-TM. Their effort resulted in a book that is an invaluable tool for researchers active in this emerging domain.

October 2014

Rachid Guerraoui
Paolo Romano

This book may be cited as
COST Action IC1001 (Euro-TM) - “Transactional Memory: Foundations, Algorithms, Tools, and Applications”, Editors Rachid Guerraoui and Paolo Romano, ISBN 978-3-319-14719-2, 2015

Table of Contents

Transactional Memory and Reliability

Distributed Transactional Memory

Applications and Self-tuning