Lecture Notes in Computer Science 9212

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

More information about this series at http://www.springer.com/series/7407

Andrzej Pelc · Alexander A. Schwarzmann (Eds.)

Stabilization, Safety, and Security of Distributed Systems

17th International Symposium, SSS 2015
Edmonton, AB, Canada, August 18–21, 2015
Proceedings

Springer

Editors
Andrzej Pelc
Université du Québec en Outaouais
Gatineau, QC
Canada

Alexander A. Schwarzmann
University of Connecticut
Storrs, CT
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-21740-6 ISBN 978-3-319-21741-3 (eBook)
DOI 10.1007/978-3-319-21741-3

Library of Congress Control Number: 2015943848

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

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

Preface

The International Symposium on Stabilization, Safety, and Security in Distributed Systems (SSS) is an international forum for researchers and practitioners working on the design and development of distributed systems that guarantee specific desired properties despite adversity, or that are able to restore the desired properties following adversarial perturbations in the computing medium building on the principles of self-stabilization. Research in distributed computing and distributed systems continues its vibrant development, marked by the importance of dynamic systems, such as peer-to-peer networks, large-scale wireless sensor networks, mobile ad hoc networks, mobile agent computing, opportunistic networks etc. Moreover, new applications such as grid and web services, banking and e-commerce, e-voting, e-health and robotics, aerospace and avionics, automotive, industrial process control, have joined the expanded landscape of distributed systems. It is becoming increasingly important to endow all such systems with built-in means for self-management, self-protection, and self-repair.

This volume contains the papers presented at the 17th International Symposium on Stabilization, Safety, and Security of Distributed Systems, held August 18–21, 2015 in Edmonton, Alberta, Canada.

This year the Program Committee was organized into several tracks reflecting most topics related to the conference interests. The tracks are: Self-Stabilization, Fault-tolerance and Dependability, Ad-hoc and Sensor Networks, Mobile Agents, System Security in Distributed Computing, and Formal Methods and Distributed Algorithms. We received 38 regular paper submissions. Each submission was reviewed by at least three Program Committee members with the help of external reviewers. Out of these 38 submissions, 16 papers were accepted for presentation at the symposium and publication in the proceedings as regular papers. The proceedings also include eight brief announcements.

Two regular papers received awards. The Best Paper Award was given to Colin Cooper, Anissa Lamani, Giovanni Viglietta, Masafumi Yamashita, and Yukiko Yamauchi for their paper "Constructing Self-Stabilizing Oscillators in Population Protocols," and the Best Student Paper Award was given to Lili Su (student) and Nitin Vaidya for their paper "Reaching Approximate Byzantine Consensus with Multi-hop Communication."

The program also included three distinguished keynote lectures by Sergio Rajsbaum (UNAM, Mexico), Roger Wattenhofer (ETH Zurich, Switzerland), and Philipp Woelfel (University of Calgary, Canada).

On behalf of the Program Committee, we thank all authors who submitted their work to SSS 2015. We gratefully acknowledge the substantial effort of the track chairs and the Program Committee members invested in paper selection. Thanks are also due to the external reviewers for their valuable and insightful comments. We also thank to the Steering Committee members for their valuable advice and guidance and to the

Organizing Committee members for their work in ensuring a successful and pleasant meeting.

Colocated with the symposium was the Summer School on Distributed Computing and Cryptography organized by Shlomi Dolev.

SSS 2015 acknowledges with gratitude the support of the Faculty of Science, University of Alberta, and EasyChair.org for the use of their system in handling submissions, managing the review process, and helping compile these proceedings.

August 2015

Andrzej Pelc
Alexander A. Schwarzmann

Organization

General Co-Chairs

Ted Herman	University of Iowa, USA
Jared Saia	University of New Mexico, USA

Program Committee Co-Chairs

Andrzej Pelc	Université du Québec en Outaouais, Canada
Alexander A. Schwarzmann	University of Connecticut, USA

Program Committee

Self-Stabilization Track

Joffroy Beauquier, *Track Chair*	Université Paris-Sud, France
Janna Burman	Université Paris-Sud, France
Ajoy Datta	University of Nevada Las Vegas, USA
Swan Dubois	Université Paris 6, France
Sukumar Ghosh	University of Iowa, USA
Shay Kutten	Technion, Israel
Christian Scheideler	University of Paderborn, Germany
Masafumi Yamashita	Kyushu University, Japan

Fault-tolerance and Dependability Track

Nitin Vaidya, *Track Chair*	University of Illinois at Urbana-Champaign, USA
Mostefaoui Achour	Université de Nantes, France
James Aspnes	Yale University, USA
Bernadette Charron-Bost	Ecole Polytechnique, France
Xavier Defago	Japan Advanced Institute of Science and Technology
Borzoo Bonakdarpour	McMaster University, Canada
Shlomi Dolev	Ben-Gurion University of the Negev, Israel
Maurice Herlihy	Brown University, USA
Vijay Garg	University of Texas at Austin, USA
Rachid Guerraoui	EPFL, Switzerland
Luis Rodrigues	Universidade de Lisboa, Portugal
Srikanth Sastry	Google, USA
Sebastien Tixeuil	Université Pierre et Marie Curie, France
Jennifer L. Welch	Texas A&M University, USA

Ad-hoc and Sensor Networks, Mobile Agents Track

Paola Flocchini, *Track Chair*	University of Ottawa, Canada
Jeremie Chalopin	CNRS/Aix-Marseille Université, France
Sandor Fekete	Technische Universität, Braunschweig, Germany
Magns M. Halldorsson	Reykjavik University, Iceland
Taisuke Izumi	Nagoya Institute of Technology, Japan
Adrian Kosowski	INRIA Bordeaux, France
Flaminia Luccio	University of Venice, Italy
Russ Martin	University of Liverpool, UK
Lata Narayanan	Concordia University, Canada
Calvin Newport	Georgetown University, USA
Koichi Wada	Hosei University, Japan

System Security in Distributed Computing Track

Alexander Russell, *Track Chair*	University of Connecticut, USA
Mohamed Gouda	University of Texas, Austin, USA
Aggelos Kiayias	University of Athens, Greece
Nicolas Nicolaou	IMDEA Networks Institute, Spain
Ravi Sundaram	Northeastern University, USA
Hong-Sheng Zhou	Virginia Commonwealth University, USA

Formal Methods and Distributed Algorithms Track

Helmut Veith, *Track Chair*	Vienna University of Technology, Austria
Parosh Abdullah	Uppsala University, Sweden
Borzoo Bonakdarpour	McMaster University, Canada
Sagar Chaki	Carnegie Mellon University, USA
Giorgio Delzanno	University of Genoa, Italy
Cezara Dragoi	INRIA, France
Pierre Ganty	IMDEA Software Institute, Spain
Swen Jacobs	Universitat des Saarlandes, Germany
Zachary Kincaid	University of Toronto, Canada
Igor Konnov	Vienna University of Technology, Austria
Ken McMillan	Microsoft Research, USA
Stefan Merz	INRIA Nancy/LORIA, France
Andreas Podelski	Universität Freiburg, Germany
Lenore D. Zuck	University of Illinois at Chicago, USA

Symposium Organization

Local Arrangements Chair

Ioanis Nikolaidis	University of Alberta, Canada

Finance Co-Chairs

Borzoo Bonakdarpour	McMaster University, Canada
H. James Hoover	University of Alberta, Canada

Publicity Chair

Maxwell Young	Drexel University, USA

Steering Committee

Anish Arora	Ohio State University, USA
Ajoy K. Datta	University of Nevada, USA
Shlomi Dolev, *Chair*	Ben-Gurion University of the Negev, Israel
Sukumar Ghosh	University of Iowa, USA
Mohamed Gouda	University of Texas at Austin, USA
Ted Herman	University of Iowa, USA
Toshimitsu Masuzawa	Osaka University, Japan
Vincent Villain	Université de Picardie Jules Verne (UPJV), France

External Reviewers

Andrew Berns
Stéphane Devismes
Anaïs Durand
Martina Eikel
Chryssis Georgiou
Yoshiaki Katayama
Shuji Kijima
Andreas Koutsopoulos
Anissa Lamani
Hammurabi Mendes
Fukuhito Ooshita
Franck Petit
Maria Potop-Butucaru
Othmane Rezine
Alexander Setzer
Devan Sohier
Thim Strothmann
Giovanni Viglietta
Bingsheng Zhang

Keynote Lectures

Distributed Runtime Verification

where combinatorics, fault-tolerance and formal methods meet

Sergio Rajsbaum

Instituto de Matemáticas, Universidad Nacional Autónoma de México,
D.F. 04510, Mexico

Abstract of Keynote Lecture

Runtime verification. RV techniques are concerned with monitoring software and hardware system executions. They are complementary, and sometimes more versatile than conventional testing, and more practical than exhaustive formal verification, such as model checking and theorem proving, as well as incomplete solutions such as testing and debugging. There is an international conference, *RV* dedicated to these techniques.

Distributed runtime verification. This talk gives an overview of distributed runtime verification (DRV). Building a decentralized runtime monitor for a distributed system is an especially difficult task since it involves designing a distributed algorithm that coordinates the monitors in order for them to reason consistently about the temporal behavior of the system. DRV techniques are less developed; they involve designing a distributed algorithm that monitors another distributed algorithm.

In an asynchronous system where processes may crash, it is impossible for the monitors to agree on the order of events in the system, due to the impossibility of solving consensus. Thus, it is unavoidable that monitors emit different opinions about the validity of the computation, that nevertheless, should be consistent with each other. Lower and upper bounds on the number of opinions that may have to be emitted, can be derived, as a function of the specification φ that is being monitored.

At the crossroads where distributed algorithms and formal methods meet. An overview of the different types of techniques used in DRV is presented, which range from formal methods techniques related to LTL and multi-valued logics, on the one hand, to algorithmic techniques related to computing snapshots in an efficient manner to reason about temporal properties of a distributed system, on the other hand, and passing through combinatorial and topological techniques. RV is an exemplary area for interdisciplinary research opportunities, given that logic and algorithmic techniques converge, and few papers explore the difficulties introduced when failures and asynchrony can occur in the system.

Supported by a UNAM-PAPIIT Grant.

Acknowledgements. The results presented involve joint work with Borzoo Bonakdarpour, Pierre Fraigniaud, Matthieu Roy, David Rosenblueth and Corentin Travers. Some of them have been published in DISC'11, OPODIS'14, RV'14, and *Distributed Computing* (2013).

A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels

Christian Decker and Roger Wattenhofer

Distributed Computing Group, ETH Zurich
{cdecker,wattenhofer}@ethz.ch

Abstract. Bitcoin does not scale, because its synchronization mechanism, the blockchain, limits the maximum rate of transactions the network can process. However, using off-blockchain transactions it is possible to create long-lived channels over which an arbitrary number of transfers can be processed locally between two users, without any burden to the Bitcoin network. These channels may form a network of payment service providers (PSP) and payments can be routed between any two users in real time, without any confirmation delay. In this work we present a protocol for duplex micropayment channels, which guarantee end-to-end security and allow instant transfers, laying the foundation of the PSP network.

Correctness Conditions for Randomized Shared Memory Algorithms

Philipp Woelfel

Department of Computer Science, University of Calgary, Canada

Abstract of Keynote Lecture

In an asynchronous shared memory system, processes communicate by applying operations on shared *base* objects. From an algorithm designer's perspective it is ideal if the operations on these objects are *atomic*, meaning that each such operation happens instantaneously. However, objects provided by systems are typically not truly atomic, and neither are objects implemented from base objects. As a result, if multiple processes concurrently execute methods on such objects, the set of all possible outcomes is difficult to predict.

For almost two decades, *linearizability*, defined by Herlihy and Wing [4], has been the gold standard among correctness conditions for non-atomic objects. It guarantees that any possible result that can arise from an interleaving of processes using linearizable operations could arise if the operations were atomic. Hence, the worst-case behaviour of algorithms can be analyzed under the assumption that all operations are atomic, even when they are only linearizable. For that reason, the terms linearizability and atomicity have often been used interchangeably (see for example [5]).

Golab, Higham, and Woelfel [2] observed that linearizable implementations do not preserve the probability distribution of the possible results if we replace atomic objects used in a *randomized* algorithm with implemented ones. An *adversary*, which schedules process steps, can "stretch out" a method call that was originally an atomic operation, and inspect the outcome of other processes coin flips before allowing the method call to be completed. As a result, replacing an atomic object with a linearizable one in a randomized algorithm amounts to increasing the power of the adversary. In order to be able to employ the power of randomization in shared memory algorithms, we need to devise new correctness conditions that eliminate the deficiencies of linearizability. In this talk the state of the art [1–3] of finding such correctness conditions will be presented.

References

1. Denysyuk, O., Woelfel, P.: Wait-freedom is harder than lock-freedom under strong linearizability (2015, submitted)
2. Golab, W., Higham, L., Woelfel, P.: Linearizable implementations do not suffice for randomized distributed computation. In: Proceedings of 43rd ACM STOC, pp. 373–382 (2011)
3. Helmi, M., Higham, L., Woelfel, P.: Strongly linearizable implementations: possibilities and impossibilities. In: Proceedings of 31st PODC, pp. 385–394 (2012)

4. Herlihy, M., Wing, J.: Linearizability: a correctness condition for concurrent objects. ACM Trans. Prog. Lang. Syst. **12**, 463–492 (1990)
5. Lynch, N.: Distributed Algorithms. Morgan Kaufmann Publishers Inc. (1996)

Contents