Tatjana Welzer · Johann Eder ·
Vili Podgorelec · Robert Wrembel ·
Mirjana Ivanović · Johann Gamper ·
Mikołaj Morzy · Theodoros Tzouramanis ·
Jérôme Darmont · Aida Kamišalić Latifić (Eds.)

Communications in Computer and Information Science 1064

New Trends in Databases and Information Systems

ADBIS 2019 Short Papers, Workshops
BBIGAP, QAUCA, SemBDM, SIMPDA, M2P, MADEISD
and Doctoral Consortium
Bled, Slovenia, September, 8–11, 2019, Proceedings

Springer

Communications in Computer and Information Science 1064

Commenced Publication in 2007

More information about this series at http://www.springer.com/series/7899

Tatjana Welzer · Johann Eder ·
Vili Podgorelec · Robert Wrembel ·
Mirjana Ivanović · Johann Gamper ·
Mikołaj Morzy · Theodoros Tzouramanis ·
Jérôme Darmont · Aida Kamišalić Latifić (Eds.)

New Trends in Databases and Information Systems

ADBIS 2019 Short Papers, Workshops
BBIGAP, QAUCA, SemBDM, SIMPDA, M2P, MADEISD
and Doctoral Consortium
Bled, Slovenia, September 8–11, 2019
Proceedings

Springer

Editors
Tatjana Welzer
University of Maribor
Maribor, Slovenia

Vili Podgorelec
University of Maribor
Maribor, Slovenia

Mirjana Ivanović
University of Novi Sad
Novi Sad, Serbia

Mikołaj Morzy
Poznań University of Technology
Poznan, Poland

Jérôme Darmont
Université Lumière Lyon 2
Lyon, France

Johann Eder
Alpen Adria University Klagenfurt
Klagenfurt am Wörthersee, Austria

Robert Wrembel
Poznan University of Technology
Poznan, Poland

Johann Gamper
Free University of Bozen-Bolzano
Bolzano, Italy

Theodoros Tzouramanis
University of Thessaly
Lamia, Greece

Aida Kamišalić Latifić
University of Maribor
Maribor, Slovenia

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-030-30277-1 ISBN 978-3-030-30278-8 (eBook)
https://doi.org/10.1007/978-3-030-30278-8

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

Preface

The European Conference on Advances in Databases and Information Systems (ADBIS) celebrated its 23rd anniversary. Previous ADBIS conferences were held in St. Petersburg (1997), Poznan (1998), Maribor (1999), Prague (2000), Vilnius (2001), Bratislava (2002), Dresden (2003), Budapest (2004), Tallinn (2005), Thessaloniki (2006), Varna (2007), Pori (2008), Riga (2009), Novi Sad (2010), Vienna (2011), Poznan (2012), Genoa (2013), Ohrid (2014), Poitiers (2015), Prague (2016), Nicosia (2017), and Budapest (2018). After 20 years the conference returned to Slovenia, and was organized in Bled.

ADBIS can be considered one of the most established and recognized conferences in Europe, in the broad field of databases and information systems. The conference aims at: (1) providing an international forum for presenting research achievements on database theory and practice, development of advanced DBMS technologies, and their applications; (2) promoting the interaction and collaboration between the database and information systems research communities from European countries and the rest of the world; (3) offering a forum for a less formal exchange of research ideas by means of affiliated workshops; and (4) activating young researchers from all over the world by means of a doctoral consortium.

ADBIS workshops have been associated with the ADBIS conference since 2005 and doctoral consortia have been associated since 2008. This long tradition continued this year. Therefore, the program of ADBIS 2019 included keynotes, research papers, thematic workshops, and a doctoral consortium (DC). The main conference, workshops, and DC had their own international Program Committees.

This volume contains 19 short research papers from the main conference, 31 workshops papers, and 5 DC papers, which were all presented at ADBIS 2019, held during September 8–11, 2019, in Bled, Slovenia.

The selected short papers span a wide spectrum of topics related to the ADBIS conference. Most of them are related to database and information systems technologies for advanced applications. Typical applications are text databases, streaming data, and graph processing. In addition, there are also papers covering the theory of databases.

The main conference received a total of 103 submissions. After a rigorous reviewing process 27 papers were accepted as full papers and 19 were selected as short papers for presentation and publication in this volume, giving an acceptance rate for short papers of 45%.

The following six workshops were run at ADBIS 2019:

- International Workshop on BI & Big Data Applications (BBIGAP), chaired by: Fadila Bentayeb (Université Lyon 2, France) and Omar Boussaid (Université Lyon 2, France)

- International Workshop on Qualitative Aspects of User-Centered Analytics (QAUCA), chaired by: Nicolas Labroche (Université de Tours, France), Patrick Marcel (Université de Tours, France), and Veronika Peralta (Université de Tours, France)
- A joint workshop Semantics in Big Data Management (SemBDM) and Data-Driven Process Discovery and Analysis (SIMPDA), chaired by: Paolo Ceravolo (Universita degli Studi di Milano, Italy), Florence Sedes (Toulouse Institute of Computer Science Research, France), Maria Teresa Gomez Lopez (University of Seville, Spain), and Maurice van Keulen (University of Twente, The Netherlands)
- Modelling is going to become Programming (M2P), chaired by: Ajantha Dahanayake (Lappeenranta University of Technology, Finland) and Bernhard Thalheim (Kiel University, Germany)
- Modern Approaches in Data Engineering and Information System Design (MADEISD), chaired by: Ivan Luković (University of Novi Sad, Serbia) and Slavica Kordić (University of Novi Sad, Serbia)

In total, 67 papers were submitted to these workshops, out of which 31 were selected for presentation and publication in this volume, giving an acceptance rate of 46%.

The ADBIS 2019 DC was a forum where PhD students had a chance to present their research ideas to the database research community, to receive inspiration from their peers and feedback from senior researchers, and to tie cooperation bounds. DC papers aim at describing the current status of the thesis research. The DC Committee accepted five papers that were presented at the DC. Two main topics emerged this year: big data integration and big data analytics.

ADBIS chairs would like to express their sincere gratitude to everyone who contributed to make ADBIS 2019 successful:

- All the organizers of the previous ADBIS workshops and conferences. They made ADBIS a valuable trademark and we are proud to continue their work.
- The authors, who submitted papers of high quality to the conference.
- The members of the international Program Committee for dedicating their time and expertise to assure a high-quality program.
- The members of ADBIS Steering Committee for proven trust and conferred organization of the conference.
- Springer for publishing these proceedings.
- Last but not least, to all the helping hands from the webmaster, programmers to technicians, and administration, without whom the organization of such a conference would not have been possible.

– Finally, we would like to express our special thanks to the local chair Lili Nemec Zlatolas for her continuous coordinating activities that ensured the success of ADBIS 2019.

July 2019

Tatjana Welzer
Johann Eder
Vili Podgorelec
Robert Wrembel
Mirjana Ivanović
Johann Gamper
Mikołaj Morzy
Theodoros Tzouramanis
Jérôme Darmont
Aida Kamišalić Latifić

Organization

Steering Committee Chair

Yannis Manolopoulos	Open University of Cyprus, Cyprus

Steering Committee

Ladjel Bellatreche	Laboratory of Computer Science and Automatic Control for Systems, France
Andras Benczur	Eötvös Loránd University, Hungary
Maria Bielikova	Slovak University of Technology, Slovakia
Barbara Catania	University of Genoa, Italy
Johann Eder	Alpen-Adria-Universität Klagenfurt, Austria
Theo Haerder	University of Kaiserslautern, Germany
Mirjana Ivanović	University of Novi Sad, Serbia
Hannu Jaakkola	Tampere University, Finland
Marite Kirikova	Riga Technical University, Latvia
Yannis Manolopoulos	Open University of Cyprus, Cyprus
Rainer Manthey	University of Bonn, Germany
Manuk Manukyan	Yerevan State University, Armenia
Tadeusz Morzy	Poznan University of Technology, Poland
Pavol Navrat	Slovak University of Technology, Slovakia
Boris Novikov	Saint Petersburg State University, Russia
George Angelos Papadopoulos	University of Cyprus, Cyprus
Jaroslav Pokorny	Charles University Prague, Czech Republic
Boris Rachev	Technical University of Varna, Bulgaria
Bernhard Thalheim	Kiel University, Germany
Goce Trajcevski	Iowa State University of Science and Technology, USA
Tatjana Welzer	University of Maribor, Slovenia
Robert Wrembel	Poznan University of Technology, Poland
Ester Zumpano	Università Della Calabria, Italy

Program Committee Chairs

Johann Eder	University of Klagenfurt, Austria
Vili Podgorelec	University of Maribor, Slovenia

Program Committee

Syed Sibte Raza Abidi	Dalhousie University, Canada
Bernd Amann	Sorbonne University, France

Costin Badica	University of Craiova, Romania
Marko Bajec	University of Ljubljana, Slovenia
Rodrigo Coelho Barros	Pontifical Catholic University of Rio Grande do Sul, Brazil
Andreas Behrend	University of Bonn, Germany
Ladjel Bellatreche	LIAS, ENSMA, France
András Benczür	Eötvös Loránd University, Hungary
Maria Bielikova	Slovak University of Technology in Bratislava, Slovakia
Nikos Bikakis	University of Ioannina, Greece
Zoran Bosnić	University of Ljubljana, Slovenia
Drazěn Brdjanin	University of Banja Luka, Bosnia and Herzegovina
Albertas Caplinskas	Vilnius University, Lithuania
Christos Doulkeridis	University of Piraeus, Greece
Johann Eder	University of Klagenfurt, Austria
Markus Endres	University of Passau, Germany
Werner Esswein	TU Dresden, Germany
Flavio Ferrarotti	Software Competence Centre Hagenberg, Austria
Flavius Frasincar	Erasmus University Rotterdam, The Netherlands
Jānis Grabis	Riga Technical University, Latvia
Francesco Guerra	University of Modena and Reggio Emilia, Italy
Giancarlo Guizzardi	Federal University of Espirito Santo, Brazil
Hele-Mai Haav	Tallinn University of Technology, Estonia
Theo Härder	University of Kaiserslautern, Germany
Tomáš Horváth	Eötvös Loránd University, Hungary
Marko Hölbl	University of Maribor, Slovenia
Andres Iglesias	University of Cantabria, Spain
Mirjana Ivanović	University of Novi Sad, Serbia
Hannu Jaakkola	Tampere University, Finland
Lili Jiang	Umea University, Sweden
Aida Kamišalić Latifić	University of Maribor, Slovenia
Mehmed Kantardzic	University of Louisville, USA
Dimitris Karagiannis	University of Vienna, Austria
Sašo Karakatič	University of Maribor, Slovenia
Zoubida Kedad	University of Versailles, France
Marite Kirikova	Riga Technical University, Latvia
Attila Kiss	Eötvös Loránd University, Hungary
Margita Kon-Popovska	Ss. Cyril and Methodius University in Skopje, North Macedonia
Harald Kosch	University of Passau, Germany
Michal Kratky	VSB-Technical University of Ostrava, Czech Republic
Ralf-Detlef Kutsche	TU Berlin, Germany
Julius Köpke	University of Klagenfurt, Austria
Dejan Lavbič	University of Ljubljana, Slovenia
Sebastian Link	The University of Auckland, New Zealand
Audrone Lupeikiene	Vilnius University, Lithuania

Federica Mandreoli	University of Modena, Italy
Yannis Manolopoulos	Open University of Cyprus, Cyprus
Manuk Manukyan	Yerevan State University, Armenia
Karol Matiasko	University of Žilina, Slovakia
Goran Mausă	University of Rijeka, Croatia
Bálint Molnár	Eötvös University of Budapest, Hungary
Angelo Montanari	University of Udine, Italy
Tadeusz Morzy	Poznan University of Technology, Poland
Boris Novikov	St. Petersburg University, Russia
Kjetil Nørvåg	Norwegian University of Science and Technology, Norway
Andreas Oberweis	Karlsruhe Institute of Technology, Germany
Andreas L. Opdahl	University of Bergen, Norway
Eneko Osaba	Tecnalia Research & Innovation, Spain
Odysseas Papapetrou	Eindhoven University of Technology, The Netherlands
András Pataricza	Budapest University of Technology and Economics, Czech Republic
Tomas Pitner	Masaryk University, Czech Republic
Vili Podgorelec	University of Maribor, Slovenia
Jaroslav Pokorný	Charles University Prague, Czech Republic
Giuseppe Polese	University of Salerno, Italy
Boris Rachev	Technical University of Varna, Bulgaria
Miloš Radovanović	University of Novi Sad, Serbia
Heri Ramampiaro	Norwegian University of Science and Technology, Norway
Stefano Rizzi	University of Bologna, Italy
Peter Ruppel	Technical University of Berlin, Germany
Gunter Saake	University of Magdeburg, Germany
Petr Saloun	VSB - Technical University of Ostrava, Czech Republic
José Luis Sánchez de la Rosa	University of La Laguna, Spain
Shiori Sasaki	Keio University, Japan
Kai-Uwe Sattler	TU Ilmenau, Germany
Miloš Savić	University of Novi Sad, Serbia
Timos Sellis	Swinburne University of Technology, Australia
Bela Stantic	Griffith University, Australia
Kostas Stefanidis	Tampere University, Finland
Claudia Steinberger	University of Klagenfurt, Austria
Sergey Stupnikov	Russian Academy of Sciences, Russia
Bernhard Thalheim	Kiel University, Germany
Raquel Trillo-Lado	Universidad de Zaragoza, Spain
Muhamed Turkanović	University of Maribor, Slovenia
Olegas Vasilecas	Vilnius Gediminas Technical University, Lithuania
Goran Velinov	Ss. Cyril and Methodius University, North Macedonia
Peter Vojtas	Charles University Prague, Czech Republic

Isabelle Wattiau	ESSEC and CNAM, France
Tatjana Welzer	University of Maribor, Slovenia
Robert Wrembel	Poznan University of Technology, Poland
Jaroslav Zendulka	Brno University of Technology, Czech Republic

Additional Reviewers

Nabila Berkani
Dominik Bork
Andrea Brunello
Loredana Caruccio
Stefano Cirillo
Victoria Döller
Peter Gašpar
Sandi Gec
Yong-Bin Kang
Selma Khouri
Haridimos Kondylakis
Ilya Makarov
Riccardo Martoglia
Matteo Paganelli
Marek Rychlý
Victor Sepulveda
Paolo Sottovia
Nicola Vitacolonna
Farhad Zafari

General Chair

Tatjana Welzer	University of Maribor, Slovenia

Honorary Chair

Ivan Rozman	University of Maribor, Slovenia

Proceedings Chair

Aida Kamišalić Latifić	University of Maribor, Slovenia

Workshops Chairs

Robert Wrembel	Poznan University of Technology, Poland
Mirjana Ivanović	University of Novi Sad, Serbia
Johann Gamper	Free University of Bozen-Bolzano, Italy

Doctoral Consortium Chairs

Jerome Darmont	Université Lumière Lyon 2, France
Mikolay Morzy	Poznan University of Technology, Poland
Theodoros Tzouramanis	University of the Aegean, Greece

Local Chair

Lili Nemec Zlatolas	University of Maribor, Slovenia

Organizing Committee

Marko Hölbl	University of Maribor, Slovenia
Luka Hrgarek	University of Maribor, Slovenia
Aida Kamišalić Latifić	University of Maribor, Slovenia
Marko Kompara	University of Maribor, Slovenia
Lili Nemec Zlatolas	University of Maribor, Slovenia
Tatjana Welzer	University of Maribor, Slovenia
Borut Zlatolas	University of Maribor, Slovenia

Workshops

Workshops Chairs

Robert Wrembel	Poznan University of Technology, Poland
Mirjana Ivanović	University of Novi Sad, Serbia
Johann Gamper	Free University of Bozen-Bolzano, Italy

Modelling Is Going to Become Programming – M2P

Chairs

Ajantha Dahanayake	Lappeenranta University of Technology, Finland
Bernhard Thalheim	Kiel University, Germany

Program Committee Members

Witold Abramowicz	Poznan University of Economics, Poland
Igor Fiodorov	Plekhanov Russian University of Economics, Russia
Holger Giese	University of Potsdam, Germany
Hannu Jaakkola	University of Tampere, Finland
Heinrich C. Mayr	University of Klagenfurt, Austria
John Mylopoulos	University of Toronto, Canada
Bernhard Rumpe	RWTH Aachen University, Germany
Klaus-Dieter Schewe	Zhejiang University, China
Veda Storey	Georgia State University, USA
Marina Tropmann-Frick	Hamburg University of Applied Sciences, Germany

Additional Reviewers

Manuela Dalibor	RWTH Aachen University, Germany
Nico Jansen	RWTH Aachen University, Germany
Jörg Christian Kirchhof	RWTH Aachen University, Germany
Judith Michael	RWTH Aachen University, Germany
Lukas Netz	RWTH Aachen University, Germany

Modern Approaches in Data Engineering and Information System Design – MADEISD

Chairs

Ivan Luković	University of Novi Sad, Serbia
Slavica Kordić	University of Novi Sad, Serbia

Program Committee Members

Paulo Alves	Polytechnic Institute of Bragança, Portugal
Marko Bajec	University of Ljubljana, Slovenia
Zoran Bosnić	University of Ljubljana, Slovenia
Moharram Challenger	University of Antwerp, Belgium
Boris Delibašić	University of Belgrade, Serbia
Joaõ Miguel Lobo Fernandes	University of Minho, Portugal
Krešimir Fertalj	University of Zagreb, Croatia
Krzysztof Goczyla	Gdánsk University of Technology, Poland
Ralf-Christian Härting	Aalen University, Germany
Dušan Jakovetić	University of Novi Sad, Serbia
Miklós Krész	InnoRenew CoE and University of Primorska, Slovenia
Dragan Maćoš	Beuth University of Applied Sciences Berlin, Germany
Zoran Marjanović	University of Belgrade, Serbia
Sanda Martinčić-Ipšić	University of Rijeka, Croatia
Cristian Mihaescu	University of Craiova, Romania
Nikola Obrenovć	Ecole Polytechnique Fédérale de Lausanne, Switzerland
Maxim Panov	Skolkovo Institute of Science and Technology, Moscow, Russia
Rui Humberto Pereira	Polytechnic Institute of Porto, Portugal
Aleksandar Popović	University of Montenegro, Montenegro
Patrizia Poščić	University of Rijeka, Croatia
Adam Przybylek	Gdánsk University of Technology, Poland
Sonja Ristić	University of Novi Sad, Serbia
Sergey Rykovanov	Skolkovo Institute of Science and Technology, Russia
Igor Rožanc	University of Ljubljana, Slovenia
Nikolay Skvortsov	Russian Academy of Sciences, Russia

A Joint Workshop on Semantics in Big Data Management – SemBDM and Data-Driven Process Discovery and Analysis – SIMPDA

Chairs

Paolo Ceravolo	Universita degli Studi di Milano, Italy
Florence Sedes	Toulouse Institute of Computer Science Research, France
Maria Teresa Gomez Lopez	University of Seville, Spain
Maurice van Keulen	University of Twente, The Netherlands

Program Committee Members

Pnina Soffer	University of Haifa, Israel
Kristof Böhmer	University of Vienna, Austria
Luisa Parody	Loyola University Andalusia, Spain
Gabriel Tavares	Londrina State University, Brazil
Roland Rieke	Fraunhofer Institute for Secure Information Technology, Germany
Angel Jesus Varela Vaca	University of Seville, Spain
Massimiliano de Leoni	University of Padova, Italy
Faiza Allah Bukhsh	University of Twente, The Netherlands
Robert Singer	FH JOANNEUM, Austria
Christophe Debruyne	Trinity College Dublin, Ireland
Antonia Azzini	Consorzio per il Trasferimento Tecnologico (C2T), Italy
Mirjana Pejic-Bach	University of Zagreb, Croatia
Marco Viviani	University of Milano-Bicocca, Italy
Carlos Fernandez-Llatas	Technical University of Valencia, Spain
Richard Chbeir	University of Pau and Pays de l'Adour, France
Manfred Reichert	University of Ulm, Germany
Valentina Emilia Balas	Aurel Vlaicu University of Arad, Romania
Mariangela Lazoi	University of Salento, Italy
Maria Leitner	Austrian Institute of Technology, Austria
Karima Boudaoud	University of Nice Sophia Antipolis, France
Chiara Di Francescomarino	Fondazione Bruno Kessler, Italy
Haralambos Mouratidis	University of Brighton, UK
Helen Balinsky	Hewlett Packard Laboratories, USA
Mark Strembeck	Vienna University of Economics and Business, Austria
Tamara Quaranta	40Labs, Italy
Yingqian Zhang	Eindhoven University of Technology, The Netherlands
Edgar Weippl	SBA Research, Austria

International Workshop on BI & Big Data Applications – BBIGAP

Chairs

Fadila Bentayeb	University of Lyon 2, France
Omar Boussaid	University of Lyon 2, France

Program Committee Members

Thierry Badard	Laval university of Quebec, Canada
Hassan Badir	University of Tanger, Morocco
Ladjel Bellatreche	National Engineering School for Mechanics and Aerotechnics, France
Nadjia Benblidia	University of Blida, Algeria

Sandro Bimonte	National Research Institute of Science and Technology for Environment and Agriculture, France
Azedine Boulmalkoul	University of Mohammadia, Morocco
Laurent d'Orazio	University of Rennes 1, France
Abdessamad Imine	University of Lorraine, France
Daniel Lemire	University of Quebec in Montreal, Canada
Gérald Gavin	University of Lyon, France
Rokia Missaoui	University of Quebec in Gatineau, Canada
Rim Moussa	University of Carthage, Tunisia
Abdelmounaam Rezgui	New Mexico Tech, USA
Olivier Teste	University of Toulouse, France
Gilles Zurfluh	University of Toulouse, France

International Workshop on Qualitative Aspects of User-Centered Analytics – QAUCA

Chairs

Nicolas Labroche	University of Tours, France
Patrick Marcel	University of Tours, France
Veronika Peralta	University of Tours, France

Program Committee Members

Julien Aligon	University of Toulouse, France
Ladjel Bellatreche	National Engineering School for Mechanics and Aerotechnics, France
Laure Berti-Equille	Aix-Marseille University, France
Ismael Caballero	University of Castilla-La Mancha, Spain
Silvia Chiusano	Polytechnic University of Turin, Italy
Jérôme Darmont	University of Lyon, France
Matteo Golfarelli	University of Bologna, Italy
Magdalini Eirinaki	San Jose State University, USA
Lorena Etcheverry	University of Uruguay, Uruguay
Zoubida Kedad	Versailles Saint-Quentin University, France
Natalija Kozmina	University of Latvia, Latvia
Nicolas Labroche	University of Tours, France
Marie Jeanne Lesot	Sorbonne University, France
Patrick Marcel	University of Tours, France
Adriana Marotta	University of Uruguay, Uruguay
Christophe Marsala	Sorbonne University, France
Elsa Negre	Paris Dauphine University, France
Veronika Peralta	University of Tours, France
Franck Ravat	University of Toulouse, France
Samira Si-said	National Conservatory of Arts and Craftsč, France
Alejandro Vaisman	Buenos Aires Institute of Technology, Argentina

Panos Vassiliadis	University of Ioannina, Greece
Gilles Venturini	University of Tours, France

Additional Reviewers

Ghislain Atemezing	MONDECA, France
Faten Atigui	Conservatoire National des Arts et Métiers (CNAM), France
Soufiane Mir	Orange Bank, France
Anton Dignös	Free University of Bozen-Bolzano, Italy
Marko Tkalčič	Free University of Bozen-Bolzano, Italy

Doctoral Consortium

Doctoral Consortium Chairs

Jerome Darmont	Université Lumière Lyon 2, France
Mikolay Morzy	Poznan University of Technology, Poland
Theodoros Tzouramanis	University of the Aegean, Greece

Doctoral Consortium Program Committee

Kamel Boukhalfa	USTHB, Algeria
Mahfoud Djedaini	University of Tours, France
Imen Megdiche	ISIS Albi, France
Jiefu Song	University of Toulouse, France

Contents

ADBIS 2019 Short Papers

ADBIS 2019 Workshop: Modern Approaches in Data Engineering and Information System Design – MADEISD

A Blockchain-Based Decentralized Self-balancing Architecture for the Web of Things

Aleksandar Tošić[1,2(✉)], Jernej Vičič[2], and Michael Mrissa[1,2]

[1] InnoRenew CoE, Livade 6, 6310 Izola, Slovenia
{aleksandar.tosic,michael.mrissa}@innorenew.eu
[2] Faculty of Mathematics, Natural Sciences and Information Technology, University of Primorska, Glagoljaška ulica 8, 6000 Koper, Slovenia
{aleksandar.tosic,jernej.vicic,michael.mrissa}@famnit.upr.si

Abstract. Edge computing is a distributed computing paradigm that relies on the computational resources of end devices in a network to bring benefits such as low bandwidth utilization, responsiveness, scalability and privacy preservation. Applications range from large scale sensor networks to IoT, and concern multiple domains (agriculture, supply chain, medicine, etc.). However, resource usage optimization is a challenge due to the limited capacity of edge devices and is typically handled in a centralized way, which remains an important limitation. In this paper, we propose a decentralized approach that relies on a combination of blockchain and a consensus algorithm to monitor network resources and, if necessary, migrate applications at run-time. We integrate our solution into an application container platform, thus providing an edge architecture capable of general purpose computation. We validate and evaluate our solution with a proof-of-concept implementation in a national cultural heritage building.

Keywords: Edge computing · Internet of Things · Decentralized applications · Blockchain

1 Introduction

In the last few years, edge computing has received a lot of attention as an alternative to cloud computing, due to the multiple advantages it offers, such as low bandwidth usage, responsiveness, scalability [10], and privacy preservation [17]. Edge computing has becomes possible due to the evolution of devices that offer more computational power than ever. Combined with application container platforms such as Docker [3] that mask heterogeneity problems, it becomes possible for connected devices to form a homogeneous distributed run-time environment. Additionally, orchestration engines (i.e., Kubernetes[1]) have been developed that

[1] https://kubernetes.io/.

T. Welzer et al. (Eds.): ADBIS 2019, CCIS 1064, pp. 325–336, 2019.
https://doi.org/10.1007/978-3-030-30278-8_34

manage and optimize usage of network, memory, storage, or processing power for edge devices and improve the global efficiency, scalability and energy management of edge platforms. However, such solutions are centralized, which means that they represent a single point of failure (SPOF), which entails several drawbacks, such as lack of reliability and security. The problem is so critical that developments for high availability have been explored, for instance with Kubernetes[2].

This paper proposes a solution that uses a decentralized algorithm that monitors network resources to drive application execution to address this problem. Our solution relies on an original combination of blockchain, a consensus algorithm, and a containerized monitoring application to enable run-time migration of applications, when relevant, according to the network state. It provides several advantages, such as verifiable optimal usage of all devices on the network, better resilience to disconnection, independence from cloud connection, improved privacy and security.

The remainder of this paper is organized in 7 sections. Section 2 introduces our motivating scenario related to a cultural heritage building and shows the need for a decentralized approach. Section 3 overviews relevant related work and highlights the originality of our approach. Section 4 details our proposed architecture and shows how it drives run-time migration of applications on the edge. Section 4.2 presents our network monitoring application and shows how monitoring takes place. In Sect. 5, we propose a technical implementation, and we validate and evaluate our solution with a proof-of-concept prototype related to our cultural heritage scenario. Section 6 discusses the results obtained and gives insights for possible future work.

2 Motivating Scenario

In this section, we illustrate the relevance of our approach with a scenario related to a Slovenian cultural heritage building located in Bled, Slovenia. This building has been equipped with multiple sensors to monitor its dynamic environment that affects the building and its contents. The collected data includes temperature, CO2, relative humidity, Volatile Organic Compounds (VOC), ambient light and atmospheric pressure. In this scenario, the following constraints motivate the need for a fully decentralized edge computing approach:

- Privacy: collected data about the state of the technological solution being deployed is classified as sensitive information. Although data about the building could be sent to the cloud, data about the state of resources needs to remain local and only accessible for administration purpose and for the deployed solution to self-manage.
- Reliability: centralized orchestration is not appropriate as data collection needs to be resilient to failure of any device. The network of devices needs to adjust to device disconnection at any time and keep operating in an optimal way.

[2] https://kubernetes.io/docs/setup/independent/setup-ha-etcd-with-kubeadm.

- Cost: reducing the overall cost by avoiding investing in a cloud infrastructure that involves monthly payments and permanent connection to maintain.
- Scalability: as the number of devices will evolve over time, it is necessary for the solution to be able to adjust to changes and homogeneously spread the computation over the network.
- Performance: reactivity to external events is improved if processing is performed on-site.
- Cost effectiveness: using existing devices that control sensors to perform necessary processing reduces the resource requirements of cloud based solutions, which reduces cost.

In this context, it is relevant to equip devices with the capacity to run applications locally and to self-manage the global network load and distribute it over connected devices, according to the state of the network. In the next section, we present related work and show the need for a decentralized self-managed platform on the edge. We also overview existing solutions to abstract from platform heterogeneity and justify the technological choice of a container platform to support our solution.

3 Background Knowledge and Related Work

A recent study by Taherizadeh et al. [19] shows that no widely-used cloud monitoring tools yet provide an integrated monitoring solution within edge computing frameworks, as some monitoring requirements have not been thoroughly met by any of them. Diallo et al. [6] present AutoMigrate, which incorporates a selection algorithm for deciding what services to migrate that maximizes the availability of migration. The system addresses most of the problems that are discussed in our paper. However, it relies on a single agent to manage services introducing a Single Point Of Failure (SPOF). The most notable difference in our implementation is a decentralized architecture that eliminates the SPOF.

3.1 Choreography Solutions for Edge Computing

Strictly observing the definition of orchestration, it always represents control from one party's perspective. This differs from choreography, which is more collaborative and allows each involved party to describe its part in the interaction [16]. However, to the authors' knowledge, there are no choreography solutions that tackle the problems defined in the previous section. Existing orchestration solutions typically rely on a master/slave model where a node is put in charge of the network and decides to allocate applications to nodes according to an optimization algorithm.

Containers as used in the purpose of this paper are run as a group of namespaced processes within an operating system, avoiding the overhead of starting and maintaining virtual machines (at the same time providing most of the functionalities).

The selected platform for our research was Docker [3] as it is the most widely used platform and one of the few that can migrate apps at runtime and enables easy communication. The migration is done by pausing the container, dumping the context of the paused container, transferring the context on a different host that can resume the execution given the context.

3.2 Decentralized Self-managing IoT Architectures

Kubernetes [8] is the most widely used orchestration tool, it is the go-to tool for orchestration in the Google cloud, and is the most used in the Microsoft Azure platform and similar products. It is also the most feature-filled orchestration tool available [12]. It has strong community support across many different cloud platforms (in addition to Google cloud, OpenStack, AWS, Azure).

AWS Elastic Container Service (AWS ECS) [1], Amazon's native container orchestration tool, is the best option for orchestration of AWS services as it is fully integrated into the Amazon ecosystem. It thus integrates easily with other AWS tools. The biggest limitation is that it is limited to Amazon services.

Docker Swarm[3] ships directly with Docker (integrates with Docker-compose) and is supposed to have the simplest configuration. However, it lacks some advanced monitoring options as compared to other products like Kubernetes.

Apache Mesos' based DC/OS[4] is a "distributed operation system" running on private and public cloud infrastructure that abstracts the resources of a cluster of machines and provides common services.

All presented architectures still have a common flaw: single point of failure and a lack of integration with edge computing.

There have been some proposed solutions that enable fully decentralized self-managing architectures for the IoT. For example, [11] focuses on a decentralized solution for energy management in IoT architectures connected to smart power grids. In [7], the authors propose a distributed IoT approach for electrical power demand management problems based on "distributed intelligence" rather than "traditional centralized control," with the system improving on many levels. Suzdalenko et al. [18] further develop the former approach by creating a decentralized distributed model of an IoT; where consumers can freely join and leave the system automatically at any time. Niyato et al. [13] present a system that uses machine-to-machine (M2M) communication to reduce the cost of a home energy management system. A distributed and decentralized microscopic simulation that eliminates the central entity and thus avoids the bottleneck in synchronization is presented in dSUMO [4]. In [2], the authors demonstrate the effectiveness of utilizing a publish/subscribe messaging model as connection means for indoor localization utilizing Wireless Sensor Networks (WSNs) through a middle-ware, the results showed that RSS reaches an acceptable level of accuracy for multiple types of applications.

[3] https://github.com/docker/swarm.
[4] https://dcos.io/.

However, all the aforementioned contributions are different from the solution we propose in this paper, at two levels. First, they mostly focus on a single specific aspect and find an optimal solution for it, without considering the fact that an IoT architecture involves multiple criteria that require optimization. In our work, we already consider multiple criteria to optimize application migration, while envisioning that this number of criteria can increase in the future. Second, as far as we know, there is no approach that combines a blockchain data structure with a consensus algorithm in a single framework with the objective to drive application migration at run-time on the edge, which is the main contribution of this paper.

4 A Decentralized Self-managing Architecture

In the following, we describe the general architecture that support our edge computing platform. Devices on the edge are nodes running node software and containerization software. A node can join the network by following a network protocol for exchanging known nodes and participate by executing the consensus algorithm. Nodes keep discovering the network by asking connected nodes for peers. For the sake of simplicity, in this paper we consider that the number of nodes remains reasonably limited, so that large scale discovery issues remain out of the scope of this paper.

4.1 General Architecture

Our devices are equipped to allow a specific containerized application (called node app) to introspect the state of the node and handle the diffusion of this information over the network. It also is responsible for maintaining the information about the other nodes up to date, for participating in the consensus algorithm, and for listening to messages coming from the exposed node API.

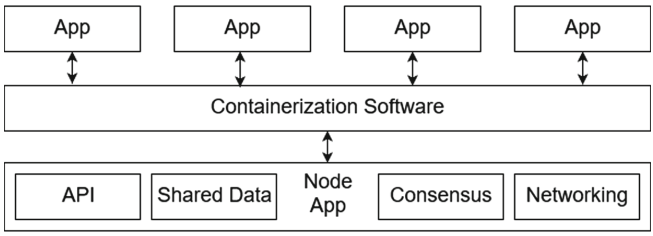


Fig. 1. Architecture of an edge device software platform: a Node App that deals with the consensus algorithm, accesses shared data and exposes the querying API is deployed into the container (in our case Docker).

Figure 1 shows the key components of Nodes in the system. The node software is deployed into the container, in our case Docker. The container mounts a direct

socket to the containerization service for querying the state of the system and managing local containers. Docker is useful here to alleviate from the typical heterogeneity problems encountered in the IoT world (different processors and OSes).

4.2 Node Application

Every 500 ms, each device collects information about the state of its neighbours. Typically, a state is a vector of scores that describes the device state and the applications being executed by the node. In this work we define a state to be a matrix of vectors

$$S\ (APP,\ CPU,\ RAM,\ DISK,\ NETWORK,\ TIMESTAMP)$$

where each vector represents an application being executed by the node and the corresponding resource consumption. Resources are reported as a fraction of the total available. In order to have comparable values between nodes, reporting on CPU usage and network utilization the CPU is normalized with the number of cores whereas network bandwidth (download/upload) is measured when transferring containers between nodes.

Monitoring resources within the P2P network is done by having nodes maintain a list of scores of all other nodes (neighbours or not). All nodes periodically send digitally signed messages containing their score to all neighbour nodes. All nodes follow simple P2P broadcasting rules that guarantee finality and efficiency in message propagation.

- If elapsed time greater then ΔST, send signed a message containing own score to all neighbour nodes.
- When receiving a new score message, check if the message was received before (compare digital signatures).
- If the message was not seen before, send it to all connected nodes with the exception of the originating node.

Where ΔST is the time interval in which the container statistics are collected and it is configurable and should depend on the time interval of the consensus algorithm. The score pool hence contains scores of all nodes participating in the network. Each score has a corresponding time-stamp which is later used by elected nodes to create a migration strategy.

Messages containing blocks can become relatively large when the number of applications in the system increases. For improved efficiency, every score message broadcast is prefaced with a "Do you need this" (DYNT) message coupled with the digital signature of the message only. Messages are sent to nodes that reply to the DYNT message to minimize bandwidth use.

Consensus Algorithm. The network requires a consensus algorithm to avoid race conditions when migrating applications. The choice of a consensus algorithm

depends on the requirements of the implementation and domain of application. In general, any consensus based on leader election can be plugged in. Examples of such consensus algorithms are Paxos [9], Raft [15], PoET [14], etc. However, in our implementation PBFT [5] was used as it is relatively simple to implement and all its properties satisfy our demands. The only real drawback of the algorithm is that the number of messages increases exponentially with the number of nodes, so it is not applicable to large networks. It was a viable alternative for our proof-of-case implementation with a limited number of nodes. The elected leader is responsible for creating a migration plan and including the resource consumption estimates in a block. The block gets digitally signed so other nodes can verify it originates from the elected leader. Nodes receiving a new block must verify the migration plan by computing it locally and comparing the results. If the migration plan is equal, they act on it, otherwise discard the block and wait for a new one. With these simple protocol rules in place the network is Byzantine fault tolerant [5]. The block verification step is necessary to minimize accidental network forks. A migration strategy is analogous to blocks in block-chain based systems. Blocks contain all the data shared among nodes in the network and include a digital signature of the previous block thus creating a block chain. In order to create a digital signature of block $n+1$ a node needs to have the digital signature of node n. A well formed block can be verified by other nodes that also have block n. In case of a malformed block, verification will fail, and nodes will reject the block, thus forcing the nodes to agree on the shared data. The block serves as an instruction set mapping applications to nodes. Consider a case with 4 nodes in set N denoted by A, B, C, and D respectively. All nodes share their score and keep a local copy of reported scores of other nodes. Each node stores a vector of applications $v \in V$ that need to be executed. Each node has a canonical list of block B of size k where k is the current block height. Table 1 shows an example of a block k which assigns every $v \in V$ to a node $n \in N$ To create block $k+1$ a node elected as leader computes an assignment such that the use of resources is optimized (improved). The input to the Algorithm 1 is limited to block data to ensure determinism that can enforce consensus. The Algorithm 1 depends on the application domain and exploring available possibilities will be subject to future work. In this paper, we use the simple described in Algorithm 1, which is deterministic and can only take the block data as input for computation. Once a block is created, currently reported scores are included that will be used to compute block $k+2$. Additionally, blocks are equipped with meta-data like block hash, previous block hash, etc. to facilitate their utilization.

5 Implementation and Evaluation

5.1 Technical Implementation

As described in Sect. 2, we have implemented and evaluated our solution with a set of sensors deployed in the cultural heritage building Mrakova Domačija in Bled, Slovenia. Each sensor is connected to a Raspberry Pi device that hosts a Linux Alpine OS in a Docker container. The container has access to the docker

```
Data: BlockData
Result: Migration plan
Max ← FindMaxLoadedNode(BlockData);
Min ← FindMinLoadedNode(BlockData);
if !AppQueue.isEmpty() then
    while !AppQueue.isEmpty() do
        Min ← FindMinLoadedNode(BlockData);
        Min.addApp(AppQueue.dequeue());
    end
else
    AppToMigrate ← Max.MaxLoadApp;
    CurrentDeltaScore ← (Max.score − Min.score);
    FutureDeltaScore ← (Max.score − AppToMigrate.score) −
    (Min.score + AppToMigrate.score);
    if Math.abs(CurrentDeltaScore > FutureDeltaScore) then
        Migrate AppToMigrate to Min;
    end
end
```

Algorithm 1. Deterministic migration plan generation algorithm

Table 1. Block data

V	Node	RAM	DISK	CPU	Average latency
v_0	A	50%	23%	90%	23 ms
v_1	B	47%	87%	23%	33 ms
v_2	C	12%	25%	15%	51 ms
v_3	A	35%	14%	56%	101 ms
v_4	D	25%	74%	16%	9 ms

daemon via unix socket. We developed our node application inside a container, it relies on the Docker introspection capacity (`docker stats` command called from our Java program) to collect information about each device. The devices simply collect temperature and relative humidity measurements and calculate their averages. It also hosts a HTTP server[5] that exposes a RESTful API providing access to the system. In such a decentralized system, interaction can be done by any node in the network as follows:

– HTTP GET gives a representation of the target node, which includes information about the state of the device as well as all the necessary information about the node (i.e., last connection time, average connection time, etc.). HTTP GET enables users to view the shared pool of resource stats nodes maintain. Most importantly, it gives a list of all applications in the system.

[5] Please note that CoAP could be used for energy saving purposes.

– HTTP PUT/POST enables users to queue an application to be run by the system.
– HTTP DELETE is utilized when an application must be deleted from the queue.

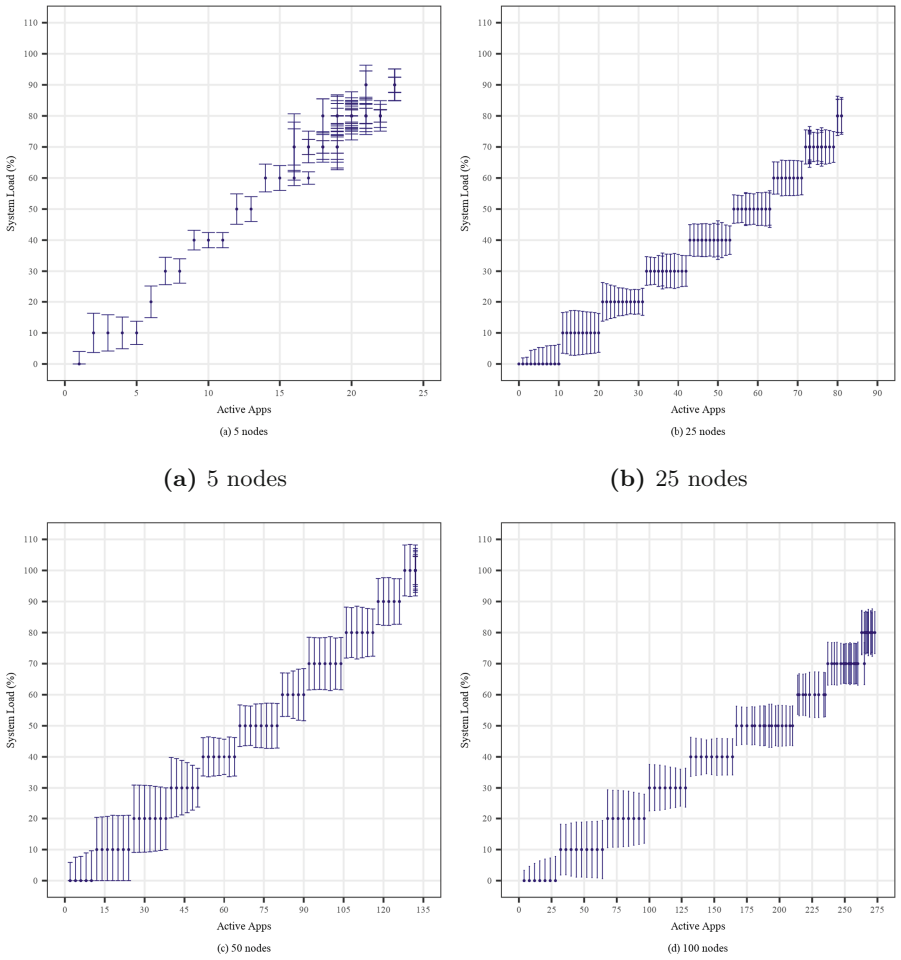
In order to deploy our prototype, we use 5 Raspberry Pi 3 Model B+ connected to Arduino Nano via USB (Universal Serial Bus), the Arduino is connected to the sensors via UART (Universal Asynchronous Receiver Transmitter) ports. We have connected DHT22 sensors to the Arduino boards to capture temperature and humidity.

5.2 Validation and Evaluation

To validate the feasibility of our approach and test its scalability we ran performance simulation test cases. In each test case, a fixed number of nodes formed a P2P network. Nodes were assigned applications to execute. Each application had a random execution time and preset resource consumption expressed in fractions between 5%–40%. For the sake of simplicity, only one resource was used (CPU). The simulation ran for 100 blocks with a block time of 1 s. Applications were queued until the average load of the entire system rose above 90%. The migration strategy was implemented based on the algorithm described in Sect. 4.2. Applications arrived in the queue with certain probability, which was gradually increased with the number of nodes in the system. From the reported resource loads of nodes (reported in %), we compute the standard deviation as a measure of how balanced resource consumption is.

In Fig. 2, we observe that the standard deviation remains low even when the number of applications in the system grows. The lower load cases where we can observe higher swings in standard deviations are expected due to the low number of applications in the system. The crossover happens when the number of applications exceeds the number of nodes and migrations can be beneficial. Below the threshold, there are bound to be nodes that do not run any applications. We can observe from Fig. 2a that as the number of nodes is low, resource balancing between nodes is effective earlier, which explains why the measures are less marked than with the other figures, that correspond to test cases where it takes the simulation a longer time to reach the point of crossover where a higher number of applications is distributed over a lower number of nodes.

Figures 2b–d show that the architecture can scale with the growing number of nodes in the network. Additionally, the naive algorithm for creating a migration strategy performed well in distributing load across the system.



(a) 5 nodes **(b)** 25 nodes

(c) 50 nodes **(d)** 100 nodes

Fig. 2. Simulation results, error bars are standard deviation of the system load

6 Discussion and Conclusion

In this paper, we propose a decentralized solution to the resource usage optimization problem, a typical issue in edge computing. Our solution avoids the single point of failure that centralized architectures suffer from and improves network resilience as it does not depend on a master node. To design our solution, we have combined a blockchain shared data structure and a consensus algorithm with a monitoring application that runs on top of the Docker platform. Such combination allows edge devices to check at run-time if there is a need for migrating an application, and to reach consensus on a decision to do so. With our

contribution, edge devices become a completely decentralized and distributed run-time platform. We have implemented and evaluated our solution with a set of sensors deployed in a cultural heritage building in Bled, Slovenia.

Results show that our approach is able to adjust and normalize the application load over a set of nodes. It also provides, thanks to the fact that the algorithm we use is deterministic and that all the data is stored in a distributed structure, the possibility to verify all the decisions that have been taken to optimize the usage of edge devices. The consensus algorithm that we use also allows adjustments to the global network behaviour for entering or leaving nodes.

Several limitations have been identified that give insights for future work. First, it is important to observe how adding and removing devices affects network behaviour and to explore how scalable our approach is over a large number of devices. Second, it seems appropriate to find out what specific aspects of use cases can help determine which consensus algorithm is most suitable for deploying our solution, in order to best match the use case requirements. Third, it includes semantically describing applications and the services that edge devices offer, to support application migration, and combine in the same architecture the need for efficiently managing network resources together with the needs of applications in terms of functionality and quality of service.

Acknowledgment. The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European regional Development Fund). The first author also acknowledges the support of the ARRS grant N1-0093.

References

1. Acuña, P.: Amazon EC2 container service. In: Acuña, P. (ed.) Deploying Rails with Docker, Kubernetes and ECS, pp. 69–98. Apress, Berkeley (2016). https://doi.org/10.1007/978-1-4842-2415-1_4
2. Al-Madani, B.M., Shahra, E.Q.: An energy aware plateform for IoT indoor tracking based on RTPS. Procedia Comput. Sci. **130**(C), 188–195 (2018)
3. Anderson, C.: Docker [software engineering]. IEEE Softw. **32**(3), 102-c3 (2015)
4. Bragard, Q., Ventresque, A., Murphy, L.: Self-balancing decentralized distributed platform for urban traffic simulation. IEEE Trans. Intell. Transp. Syst. **18**(5), 1190–1197 (2017)
5. Castro, M., Liskov, B., et al.: Practical byzantine fault tolerance. In: OSDI, vol. 99, pp. 173–186 (1999)
6. Diallo, M.H., August, M., Hallman, R., Kline, M., Slayback, S.M., Graves, C.: Automigrate: a framework for developing intelligent, self-managing cloud services with maximum availability. Cluster Comput. **20**(3), 1995–2012 (2017)
7. Higgins, N., Vyatkin, V., Nair, N.K.C., Schwarz, K.: Distributed power system automation with IEC 61850, IEC 61499, and intelligent control. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **41**(1), 81–92 (2011)
8. Hightower, K., Burns, B., Beda, J.: Kubernetes: Up and Running: Dive Into the Future of Infrastructure. O'Reilly Media, Inc., Sebastopol (2017)

9. Lamport, L., et al.: Paxos made simple. ACM SIGACT News **32**(4), 18–25 (2001)
10. Mach, P., Becvar, Z.: Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017)
11. Maior, H.A., Rao, S.: A self-governing, decentralized, extensible internet of things to share electrical power efficiently. In: 2014 IEEE International Conference on Automation Science and Engineering (CASE), pp. 37–43. IEEE (2014)
12. Medel, V., Rana, O., Bañares, J.Á., Arronategui, U.: Modelling performance & resource management in Kubernetes. In: 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), pp. 257–262. IEEE (2016)
13. Niyato, D., Xiao, L., Wang, P.: Machine-to-machine communications for home energy management system in smart grid. IEEE Commun. Mag. **49**(4), 53–59 (2011). https://doi.org/10.1109/MCOM.2011.5741146
14. Olson, K., Bowman, M., Mitchell, J., Amundson, S., Middleton, D., Montgomery, C.: Sawtooth: An Introduction. The Linux Foundation, January 2018
15. Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm. In: 2014 USENIX Annual Technical Conference (USENIX ATC 2014), pp. 305–319 (2014)
16. Peltz, C.: Web services orchestration and choreography. Computer **36**(10), 46–52 (2003)
17. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
18. Suzdalenko, A., Galkin, I.: Instantaneous, short-term and predictive long-term power balancing techniques in intelligent distribution grids. In: Camarinha-Matos, L.M., Tomic, S., Graça, P. (eds.) DoCEIS 2013. IAICT, vol. 394, pp. 343–350. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37291-9_37
19. Taherizadeh, S., Jones, A.C., Taylor, I., Zhao, Z., Stankovski, V.: Monitoring self-adaptive applications within edge computing frameworks: a state-of-the-art review. J. Syst. Softw. **136**, 19–38 (2018)

Author Index