ICAS 2021

The Seventeeth International Conference on Autonomic and Autonomous Systems

ISBN: 978-1-61208-854-9

May 30th – June 3rd, 2021

ICAS 2021 Editors

Roy Sterritt, Ulster University, UK

Mark J. Balas, Texas A&M University, USA

ICAS 2021

Foreword

The Seventeenth International Conference on Autonomic and Autonomous Systems (ICAS 2021), held between May 30 – June 3rd, 2021, was a multi-track event covering related topics on theory and practice on systems automation, autonomous systems and autonomic computing.

The main tracks referred to the general concepts of systems automation, and methodologies and techniques for designing, implementing and deploying autonomous systems. The next tracks developed around design and deployment of context-aware networks, services and applications, and the design and management of self-behavioral networks and services. We also considered monitoring, control, and management of autonomous self-aware and context-aware systems and topics dedicated to specific autonomous entities, namely, satellite systems, nomadic code systems, mobile networks, and robots. It has been recognized that modeling (in all forms this activity is known) is the fundamental for autonomous subsystems, as both managed and management entities must communicate and understand each other. Small-scale and large-scale virtualization and model-driven architecture, as well as management challenges in such architectures are considered. Autonomic features and autonomy requires a fundamental theory behind and solid control mechanisms. These topics gave credit to specific advanced practical and theoretical aspects that allow subsystem to expose complex behavior. We aimed to expose specific advancements on theory and tool in supporting advanced autonomous systems. Domain case studies (policy, mobility, survivability, privacy, etc.) and specific technology (wireless, wireline, optical, e-commerce, banking, etc.) case studies were targeted. A special track on mobile environments was indented to cover examples and aspects from mobile systems, networks, codes, and robotics.

Pervasive services and mobile computing are emerging as the next computing paradigm in which infrastructure and services are seamlessly available anywhere, anytime, and in any format. This move to a mobile and pervasive environment raises new opportunities and demands on the underlying systems. In particular, they need to be adaptive, self-adaptive, and context-aware.

Adaptive and self-management context-aware systems are difficult to create, they must be able to understand context information and dynamically change their behavior at runtime according to the context. Context information can include the user location, his preferences, his activities, the environmental conditions and the availability of computing and communication resources. Dynamic reconfiguration of the context-aware systems can generate inconsistencies as well as integrity problems, and combinatorial explosion of possible variants of these systems with a high degree of variability can introduce great complexity.

Traditionally, user interface design is a knowledge-intensive task complying with specific domains, yet being user friendly. Besides operational requirements, design recommendations refer to standards of the application domain or corporate guidelines.

Commonly, there is a set of general user interface guidelines; the challenge is due to a need for cross-team expertise. Required knowledge differs from one application domain to another, and the core knowledge is subject to constant changes and to individual perception and skills.

Passive approaches allow designers to initiate the search for information in a knowledge-database to make accessible the design information for designers during the design process. Active approaches, e.g., constraints and critics, have been also developed and tested. These mechanisms deliver information (critics) or restrict the design space (constraints) actively, according to the rules and guidelines. Active and passive approaches are usually combined to capture a useful user interface design.

We take here the opportunity to warmly thank all the members of the ICAS 2021 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICAS 2021. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICAS 2021 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICAS 2021 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the fields of autonomic and autonomous systems.

We are convinced that the participants found the event useful and communications very open.

ICAS 2021 Chairs:

ICAS 2021 Steering Committee
Roy Sterritt, Ulster University, UK
Mark J. Balas, Texas A&M University, USA
Radu Calinescu, University of York, UK
Karsten Böhm, Fachhochschule Kufstein, Austria
Jacques Malenfant, Sorbonne Université | LIP6 Lab, France
Claudius Stern, biozoom services GmbH - Kassel | FOM University of Applied Sciences – Essen, Germany
Petr Skobelev, Knowledge Genesis Group / Samara Technical University, Russia

ICAS 2021 Industry/Research Advisory Committee
Loris Penserini, Informatica e Società Digitale - IES, Italy
Stefanos Vrochidis, Centre for Research and Technology Hellas - Thermi-Thessaloniki, Greece
Tsuyoshi Ide, IBM T. J. Watson Research Center, USA
Rajat Mehrotra, Intelligent Automation Inc., USA
Andreas Kercek, Lakeside Labs GmbH, Austria
Satoshi Kurihara, University of Electro-Communications, Japan
Elisabetta Di Nitto, Politecnico di Milano, Italy
Karsten Böhm, Fachhochschule Kufstein, Austria
Richard Anthony, University of Greenwich, UK

ICAS 2021 Publicity Chairs
Daniel Basterretxea, Universitat Politecnica de Valencia, Spain
Marta Botella-Campos, Universitat Politecnica de Valencia, Spain

ICAS 2021

Committee

ICAS 2021 Steering Committee

Roy Sterritt, Ulster University, UK
Mark J. Balas, Texas A&M University, USA
Radu Calinescu, University of York, UK
Karsten Böhm, Fachhochschule Kufstein, Austria
Jacques Malenfant, Sorbonne Université | LIP6 Lab, France
Claudius Stern, biozoom services GmbH - Kassel | FOM University of Applied Sciences – Essen, Germany
Petr Skobelev, Knowledge Genesis Group / Samara Technical University, Russia

ICAS 2021 Industry/Research Advisory Committee

Loris Penserini, Informatica e Società Digitale - IES, Italy
Stefanos Vrochidis, Centre for Research and Technology Hellas - Thermi-Thessaloniki, Greece
Tsuyoshi Ide, IBM T. J. Watson Research Center, USA
Rajat Mehrotra, Intelligent Automation Inc., USA
Andreas Kercek, Lakeside Labs GmbH, Austria
Satoshi Kurihara, University of Electro-Communications, Japan
Elisabetta Di Nitto, Politecnico di Milano, Italy
Karsten Böhm, Fachhochschule Kufstein, Austria
Richard Anthony, University of Greenwich, UK

ICAS 2021 Publicity Chairs

Daniel Basterretxea, Universitat Politecnica de Valencia, Spain
Marta Botella-Campos, Universitat Politecnica de Valencia, Spain

ICAS 2021 Technical Program Committee

Lounis Adouane, Université de Technologie de Compiègne (UTC), France
Alba Amato, Institute for High-Performance Computing and Networking (ICAR), Napoli, Italy
Mark Balas, Texas A&M University, USA
Ramón Barber Castaño, Universidad Carlos III de Madrid, Spain
Malek Ben Salem, Accenture Labs, USA
Julita Bermejo-Alonso, Universidad Politécnica de Madrid (UPM), Spain
Chiara Bersani, Polytechnic School University of Genova, Italy
Navneet Bhalla, University College London, UK
Estela Bicho, University of Minho / Centre Algoritmi / CAR group, Portugal
Karsten Böhm, Fachhochschule Kufstein, Austria
Kamel Bouzgou, Université des sciences et de la technologie Oran USTO-MB, Algeria / Université Paris Saclay - Univ. Evry, France
Kenny Bowers, Georgia Tech Research Institute, USA

Estelle Bretagne, University of Picardie Jules Verne / lab MIS (modeling, information and systems), France
Ivan Buzurovic, Harvard University, Boston, USA
Radu Calinescu, University of York, UK
Paolo Campegiani, Bit4id, Italy
Valérie Camps, Paul Sabatier University - IRIT, Toulouse, France
Elisa Capello, Politecnico di Torino and CNR-IEIIT, Italy
Constantin F. Caruntu, “Gheorghe Asachi” Technical University of Iasi, Romania
Dario Cazzato, Interdisciplinary Centre for Security, Reliability and Trust | University of Luxembourg, Luxembourg
Meghan Chandarana, NASA Ames Research Center, USA
Wen-Chung Chang, National Taipei University of Technology, Taiwan
Tawfiq Chowdhury, University of Notre Dame, Indiana, USA
Stéphanie Combettes, University Paul Sabatier of Toulouse | IRIT Lab, France
Cosmin Copot, University of Antwerp, Belgium
Shreyansh Daftry, NASA Jet Propulsion Laboratory | California Institute of Technology, Pasadena, USA
Giulia De Masi, Zayed University / Rochester Institute of Technology (RIT), Dubai, UAE
Angel P. del Pobil, Jaume I University, Spain
Daniel Delgado Bellamy, University of the West of *England*, Bristol, UK
Sotirios Diamantas, Tarleton State University | Texas A&M System, USA
Manuel J. Domínguez Morales, University of Seville, Spain
Larbi Esmahi, Athabasca University, Canada
Nicola Fabiano, Studio Legale Fabiano, Italy / International Institute of Informatics and Systemics (IIIS), USA
Hugo Ferreira, INESC TEC / Porto Polytechnic Institute, Portugal
Daniel Filipe Albuquerque, Polytechnic Institute of Viseu, Portugal
Terrence P. Fries, Indiana University of Pennsylvania, USA
Wai-keung Fung, Robert Gordon University, Aberdeen, UK
Zhiwei Gao, Northumbria University, UK
Javad Ghofrani, HTW Dresden University of Applied Sciences, Germany
Martin Giese, University Clinic of Tuebingen, Germany
Philippe Giguère, Laval University, Canada
Jordi Guitart, Universitat Politècnica de Catalunya (UPC), Spain
Maki Habib, The American University in Cairo, Egypt
Cédric Herpson, University Pierre and Marie Curie (UPMC) | LIP6, Paris, France
Randa Herzallah, University of Aston, UK
Gerold Hölzl, University of Passau, Germany
Wladyslaw Homenda, Warsaw University of Technology, Poland
Wei-Chiang Hong, School of Computer Science and Technology | Jiangsu Normal University, China
Kamran Iqbal, University of Arkansas at Little Rock, USA
Konstantinos Ioannidis, Information Technologies Institute - Centre for Research and Technology Hellas, Thessaloniki, Greece
Omar Jasim, University of Sheffield, UK
Hasan Ali Khattak, National University of Sciences and Technology (NUST), Islamabad, Pakistan
Miquel Kegeleirs, IRIDIA | Université Libre de Bruxelles, Belgium
Lial Khaluf, Independent Researcher, Germany
Igor Kotenko, SPIIRAS and ITMO University, Russia
HakKeung Lam, King's College London, UK

Charles Lesire, ONERA/DTIS | University of Toulouse, France
Baoquan Li, Tiangong University, China
Guoyuan Li, Norwegian University of Science and Technology, Norway
Hsieh-Yu Li, Singapore University of Technology and Design, Singapore
Jiaoyang Li, University of Southern California, USA
Ji-Hong Li, Korea Institute of Robot and Convergence, Republic of Korea
Juncheng Li, East China Normal University, Shanghai, China
Yangmin Li, The Hong Kong Polytechnic University, Hong Kong
Enjie Liu, University of Bedfordshire, UK
Eric Lucet, CEA | LIST | Interactive Robotics Laboratory, Palaiseau, France
Jacques Malenfant, Sorbonne Université - CNRS, France
Guilhem Marcillaud, Paul Sabatier University, IRIT, France
Aurelian Marcu, Center for Advance Laser technology CETAL - National Institute for Laser Plasma and Radiation Physics, Romania
Philippe Martinet, INRIA Sophia Antipolis, France
Ignacio Martinez-Alpiste, University of the West of Scotland, UK
Rajat Mehrotra, Teradata Inc., Santa Clara,, USA
René Meier, Hochschule Luzern, Switzerland
Oliver Michler, Technical University Dresden, Germany
Sérgio Monteiro, Centro Algoritmi | University of Minho, Portugal
Naghmeh Moradpoor, Edinburgh Napier University, UK
Paulo Moura Oliveira, UTAD University, Vila Real, Portugal
Luca Muratore, Istituto Italiano di Tecnologia, Genova, Italy / The University of Manchester, UK
Taro Nakamura, Chuo University, Japan
Roberto Nardone, University of Reggio Calabria, Italy
Rafael Oliveira Vasconcelos, Federal University of Sergipe (UFS), Brazil
Flavio Oquendo, IRISA - University of South Brittany, France
Luigi Palmieri, Robert Bosch GmbH, Corporate Research, Germany
Eros Pasero, Politecnico di Turin, Italy
Timothy Patten, University of Technology Sydney, Australia
Ling Pei, Shanghai Jiao Tong University, China
Damien Pellier, Université Grenoble Alpes, France
Van-Toan Pham, National Taipei University of Technology, Taiwan
Johan Philips, KU Leuven, Belgium
Agostino Poggi, DII - University of Parma, Italy
Radu-Emil Precup, Politehnica University of Timisoara, Romania
José Ragot, Université de Lorraine, France
Leonīds Ribickis, Riga Technical University, Latvia
Douglas Rodrigues, Paulista University - UNIP, Brazil
Oliver Roesler, Vrije Universiteit Brussel, Belgium
Joerg Roth, Nuremberg Institute of Technology, Germany
Loris Roveda, Università della Svizzera italiana (USI), Switzerland
Spandan Roy, International Institute of Information Technology, Hyderabad, India
Fariba Sadri, Imperial College London, UK
Mohammad Safeea, Coimbra University, Portugal / ENSAM, Lille, France
Lakhdar Sais, CNRS | Artois University, Lens, France
Jagannathan Sarangapani, Missouri University of Science and Technology, USA
Cornelia Schulz, University of Tübingen, Germany

Vesna Sesum-Cavic, TU Wien, Austria
Abdel-Nasser Sharkawy, South Valley University, Qena, Egypt
Mahmoud Shafiee, University of Kent, Canterbury, UK
Rajat Emanuel Singh, North Carolina State University / University of North Carolina at Chapel Hill / University of Arkansas at Little Rock, USA
Inderjeet Singh, University of Texas at Arlington Research Institute (UTARI), USA
Edoardo Sinibaldi, Istituto Italiano di Tecnologia (IIT), Italy
Petr Skobelev, Samara Technical University / Knowledge Genesis Group, Russia
Mohammad Divband Soorati, University of Southampton, UK
Bernd Steinbach, University of Mining and Technology, Freiberg, Germany
Claudius Stern, FOM University of Applied Sciences, Essen, Germany
Roy Sterritt, Ulster University, UK
Yun-Hsuan Su, University of Washington, Seattle, USA
Alireza Taheri, Sharif University of Technology, Tehran, Iran
Saied Taheri, Virginia Tech, USA
Omar Tahri, PRISME | INSA Centre Val-de-Loire, France
Brahim Tamadazte, FEMTO-ST Institute / CNRS, France
Maryam Tebyani, University of California, Santa Cruz, USA
Giorgio Terracina, Università della Calabria, Italy
Carlos M. Travieso-González, Institute for Technological Development and Innovation in Communications (IDeTIC) | University of Las Palmas de Gran Canaria (ULPGC), Spain
Xuan-Tung Truong, Le Quy Don Technical University, Vietnam
Paulo Urbano, Universidade de Lisboa, Portugal
Ramon Vilanova, School of Engineering - UAB, Spain
Nikolaos (Nikos) Vitzilaios, University of South Carolina, USA
Stefanos Vrochidis, Information Technologies Institute | Centre for Research and Technology Hellas, Greece
Dingkang Wang, University of Florida, USA
Mingfeng Wang, The University of Nottingham, UK
Yin-Tien Wang, Tamkang University, Taiwan
Guowu Wei, University of Salford, UK
Hua-Liang Wei, University of Sheffield, UK
Luke Wood, University of Hertfordshire, UK
Haiyan Wu, CAS Key Laboratory of Behavioral Science | University of Chinese Academy of Sciences, China
Yuanlong Xie, Huazhong University of Science and Technology, China
Reuven Yagel, Azrieli - Jerusalem College of Engineering, Israel
Chenguang Yang, University of the West of England, Bristol, UK
Linda Yang, University of Portsmouth, UK
Haichao Zhang, Horizon Robotics, USA
Vadim Zhmud,Novosibirsk State Technical University, Russia

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Towards Elastic Edge Computing Environments: An Investigation of Adaptive Approaches

Abdullah Fawaz Aljulayfi[1, 2] and Karim Djemame[1]
[1]School of Computing
[1]University of Leeds
[1]Leeds, United Kingdom
[2]Prince Sattam Bin Abdulaziz University, Kingdom of Saudi Arabia
e-mail: {ml16afa, K.Djemame}@leeds.ac.uk

***Abstract*— The workload dynamicity of internet of things devices represents a substantial challenge for edge computing environments as it often has limited resources. It requires an efficient elasticity framework that aware of its operational environment in order to adapt in accordance to workload fluctuation which contributes towards efficient resource utilisation, high acceptance rate, and avoids quality of service violation. The edge computing elasticity can be provided through a self-adaptive system that is capable of taking the proper elasticity decisions. This self-adaptive system can be designed using a proactive-, reactive-, or hybrid-adaptation. However, the performance of these adaptation approaches may vary according to the domain, application, and workload. Therefore, this paper designs an edge computing self-adaptive system that can support proactive-, reactive-, and hybrid-adaptation. It also conducts simulation-based investigations on the performance of the adaptation approaches in an edge computing environment under different workloads and application scenarios. The experimental results reveal that the hybrid adaptation performs at least 10% better than other approaches whereas the performance of both proactive and reactive adaptations is application scenarios dependent.**

Keywords- Elasticty; Auto-scaling; Proactive; Reactive; Hybrid.

I. INTRODUCTION

The Internet of Things (IoT) [1] has become a part of everyday life. This technology provides wide-ranging benefits and is extensively used in various domains, such as healthcare and industry, for increased efficiency and productivity [2][3][4]. The massive growth of the IoT devices besides their requirements, such as low latency, location awareness, and mobility, represent a bottleneck for Cloud Computing (CC) [5]. Thus, Edge Computing (EC) considers a promising paradigm to support IoT devices by leveraging the CC resources to the edge of the network addressing their requirements. However, its workload dynamicity represents a main challenge as EC infrastructures often have limited resources [5][6][7][8]. This requires an efficient elastic resource provision framework to cope with the workload dynamicity to support resources scaling up and down in accordance to workload demand. Further, the proper elasticity decision can contribute to avoid both resources over- and under-provisioning.

An agile elasticity framework can be provided via Self-Adaptive System (SAS) which is a promising solution that provides autonomic resource management in such complex systems [9][10][11]. This allows adapting in accordance to the workload dynamicity over time. The SAS can be designed using proactive, reactive, or hybrid adaptation approaches. However, the performance of these approaches vary depending on the domain, application, and workload [12]. Therefore, the investigation of these approaches in the EC environment is an obvious research problem that aims to design an elastic SAS framework using the most appropriate adaptation approach that suites EC requirements.

Our previous work has proposed a Machine Learning Based Context-aware Prediction Framework [13]. This is extended in this paper by conducting thorough empirical investigations and evaluation of the performance of the adaptation approaches in an EC environment. These approaches are evaluated under different IoT devices' workload, application scenarios, and hypotheses.

The main contribution of this paper can be summarised as follows:

- Design an elasticity SAS framework that can support proactive-, reactive-, or hybrid-adaptation where the most proper approach can be selected. The framework itself is supported by four algorithms, namely proactive, reactive, hybrid, and admission control algorithms.
- Profile six IoT applications in a containerised edge environment which help their simulation in EC environments. This profiling considers details about applications' latency requirements, the required resources, and the uploading/ downloading data size, which are specified based on real scenarios.
- Investigate the effectiveness of the elasticity SAS in the EC environment using the adaptation approaches, which shows the suitability of these approaches to the EC. This investigation considers three real workload and two applications scenarios (i.e., mixed applications and single application) and a range of evaluation metrics, e.g., task acceptance rate and servers' utilisation. Additionally, some recommendations are made accordingly about the suitability of the SAS in EC computing environments and its design.

The rest of this paper is organised as follows. Section II discusses the related work and positions the paper. It is followed by Section III which presents the elasticity SAS framework. Section IV illustrates the experimental design. The performance of SAS is evaluated in Section V. The conclusion and future work are presented in Section VI.

II. RELATED WORK

This section presents the related work in relation to the adaptation approaches which are proactive-, reactive-, and hybrid-adaptation. Further, it positions this paper across the literature by highlighting its contributions.

A. *Proactive Adaptation*

Proactive-based SAS is a SAS that uses the collected historical data to anticipate the future system behaviour or environmental changes [14][15][16]. The future anticipation can be in different folds, such as workload and performance. The main objective of this approach is to act prior to an event occurring which helps to optimise the resource utilisation, avoid Quality of Service (QoS) violation, and support the elasticity [15][17][18].

This approach has been widely used in the literature. For instance, Spatharakis, D. et al. [19] propose a two-layers EC system architecture for location-based services which can be consumed by IoT or mobile devices. It is supported by an offloading decision mechanism and applications' performance requirement profiling mechanism. Furthermore, the Kalman Filtering estimation method is adopted for future request estimation.

In [20], an energy-aware cost prediction framework is presented using Auto Regression Integrated Moving Average (ARIMA) and Linear Regression (LR) as prediction models. The ARIMA is used for predicting Virtual Machines (VM) Central Processing Unit (CPU) utilisation, Random Access Memory-, Disk-write-, and Network-usage. The LR is utilised for predicting the physical machines' CPU utilisation. The work in [20] is extended using the same methods to propose a performance and energy-based cost prediction framework [21]. Support Vector Regression is another Machine Learning (ML) method that is adopted in [16] to introduce an auto-scaling system for web servers. Further, the queuing theory is adopted as a performance model.

An auto-scaling method for containerised micro-services in Fog Computing (FC) environment is proposed in [22]. It is driven by two ML methods, which are Decision Tree Regression and Elastic Net for learning auto-scaling policy and workload forecasting using small and large window size. A thorough evaluation is performed using both synthetic and real workload traces.

B. *Reactive Adaptation*

The reactive-based SAS is a system that monitors operational environment continuously to trigger a specific event when a condition is satisfied [14]. Although, this kind of adaptation may lead to system instability and late decision, it is the common approach found in the literature.

A multi-agent-based resource provision system architecture for FC is proposed in [23]. It aims to provide a self-adaptive and self-sustainable load-balancing system. It mainly relies on a threshold-based categorisation algorithm that arranges fog nodes based on their workload (i.e., overloaded, underloaded, and balance). The system architecture considers both fog and cloud layers where the cloud layer can be utilised in case there is a need for further processing storage, or the fog layer is fully utilised. The system is evaluated using Poisson distribution synthetic workload that is conducted on iFogSim. In [24], a container-as-service system architecture for task selection and scheduling for real-time data processing in an EC is proposed. This system migrates the containers reactively when over/under-utilisation of the servers is triggered aiming to maintain the Service Level Agreement (SLA) objectives and reduce the energy consumption.

ML methods can also be used in this approach. For example, a resource allocation agent for MEC is developed using deep reinforcement learning [25]. It aims to improve the end-to-end reliability and avoid QoS violation where the decision is made using channel quality, data packet size, and waiting time. The reactive adaptation decisions include changing the scheduling policy and adding/removing tasks.

In [26], a reactive method is introduced for allocating the web-based resource in a CC. It is performed using the user demand targeting the total deployment cost and the QoS. In [27], an energy and SLA-aware self-adaptive resource management scheme for CC is proposed. It uses size of input queue, number of available VMs, and number of provisioned VMs to adjust the number of running physical machines.

C. *Hybrid Adaptation*

A hybrid SAS uses both proactive and reactive adaptation. The reactive adaptation is used as a back-up for unpredicted occurrences [12]. The benefit of this combination can be seen in uncertain environments [28]. A limited of research is conducted considering this approach as it is complex and requires consistent decisions.

A Monitor, Analyse, Plan, and Execute over shared Knowledge (MAPE-K)-based resource provisioning framework is designed for IoT applications in a FC environment with the possibility to utilise the cloud layer [7]. It uses several statistical workload forecasting methods (e.g., Auto Regression Moving Average (ARMA) and ARIMA). An evaluation is performed to identify the most appropriate forecasting model. Further, the scaling decision of the fog nodes is made based on a Bayesian learning technique. On the reactive side, a threshold-based technique is used to make the cloud layer offloading decisions. The evaluation is conducted on iFogSim using two synthetic workloads (i.e., Smooth and bursty workloads) and New York city taxi trip real workload.

An elastic cloud platform for web applications based on Docker is designed in [29]. In terms of proactive adaptation, it uses second order ARMA for forecasting the number of requests. In terms of reactive adaptation, a resource utilisation threshold is used. The adaptation actions are adding/removing and starting/stopping the containers.

A cloud-assisted EC system architecture is proposed in [30]. It allows monitoring the EC resources by three main strategies. The first strategy is an elasticity provision which is responsible for forecasting the workload using both ARIMA and neural network. Another strategy is a dynamic replica placement across both cloud and edge layers. The last strategy is the migration of the data from the cloud to the edge once the edge workload decreases with consideration to data reliability, migration time, and cost.

The SAS is not only related to elasticity problems but can also be used to manage the power and energy consumption. For instance, an energy-aware SAS for CC is presented in [31] to minimise the power and energy consumption where adaptation changes are implemented the in the VM level.

D. Related work limitations

Although, a considerable body of research has conducted using different SAS approaches considering either CC, EC, or both environments, several limitations and clear research gaps can be identified. These limitations are related to the hybrid SAS consideration, SAS design, the suitability of SAS to the targeted environment, and the adopted workload. In fact, these limitations become obvious when considering the EC environments.

The literature limitations can be summarized as follows. Firstly, a limited research effort has been conducted using the hybrid SAS which represents a clear research gap, especially for EC environments. Another limitation is the consideration of the theoretical perspective when designing and implementing the SAS using the adaptation approaches where the adaptation approaches performance may vary based on environment, application, and workload. This means a thorough investigation about the adaptation approaches considering the EC environment, IoT applications, and workload is an open research problem and not been tackled yet. Furthermore, the use of real EC workload in the SAS evaluation represents a significant weakness due to the use of either synthetic or real cloud workload to perform the SAS evaluation in EC environments. Therefore, this paper aims to address the mentioned limitations by designing an elasticity SAS framework considering the suitability of the adaptation approach to the EC environments. Further, this design is driven via thorough investigations on the performance of proactive, reactive, and hybrid adaptation in the EC environment considering two applications' scenarios (i.e., single and mixed) and real workload.

III. PROPOSED SELF-ADAPTIVE SYSTEM

This section presents the proposed elasticity SAS framework of the EC environments.

A. MAPE-based Elasticity SAS Framework

The proposed elasticity SAS framework is designed using a MAPE-based control loop. The use of this control loop allows the system to have a full and autonomic management of the resources over time in order to act in accordance to the workload variation. This means the system can instantiate/terminate containers as well as accept/reject requests in an autonomic fashion where each request represents an IoT task that needs to be executed by containerised application. Additionally, the proposed framework is designed to support three adaptation approaches which are proactive-, reactive-, and hybrid-adaptation. The SAS implementation details are provided in Section IV-A.

In this section, the proposed SAS is discussed according to each activity in the MAPE loop, which are monitor, analyse, plan, and execute. The SAS is shown in Figure 1 and divided into regions based on MAPE activities where each activity is highlighted using different colour. These colours are red for Monitor, yellow for Analyse, green for Plan, and blue for Execute. Further, the main contributions of this paper are highlighted in grey.

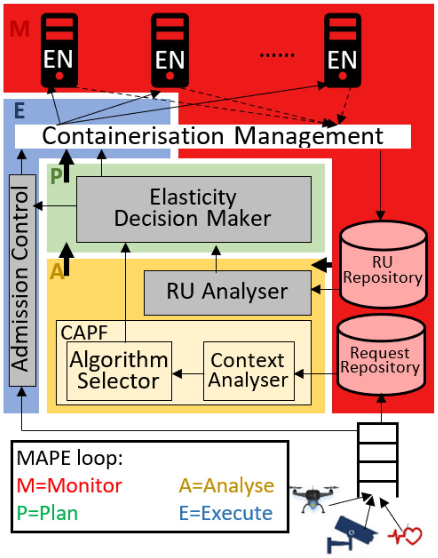


Figure 1. MAPE-based elasticity SAS framework.

1) ***Monitor:*** is responsible for collecting the data of the Edge Node (EN) resources and the IoT devices' workload. The EN resources include the number of containers which is used to instantiate/terminate containers by all adaptation approaches. This data is stored in the Resource Utilisation (RU) repository. The workload history of IoT devices is stored in the Request Repository and utilised to forecast the future IoT workload.

2) ***Analyse:*** is responsible for analysing the stored data by the monitor activity. It consists of two main components. The first component is the Context-aware Prediction Framework (CAPF) that represents the core of proactive adaptation. The CAPF is responsible for forecasting the future workload by utilising the workload history that is received from Request Repository and applying the most appropriate ML algorithm with consideration to the workload context. In this paper, this component is used as a black box as a full paper was published about CAPF in [13]. The second component is RU Analyser which is an important component for all adaptation approaches. In terms of proactive adaptation, it is responsible for analysing the run-time EN resources (i.e., number of idle containers) which can be used in the elasticity decision-making process. In case of the reactive adaptation, it is responsible for triggering the number of containers when it is below the threshold.

3) ***Plan:*** is responsible for making elasticity decision (i.e., instantiate/ terminate) where the action is taken by the Elasticity Decision Maker. In case of the proactive adaptation, this component uses both the forecasting results that are generated by CAPF and the RU analyser to make the elasticity decision. In case of the reactive adaptation, it only utilises the RU analyser results to make the elasticity decision. Lastly, in the hybrid adaptation, both CAPF and RU analyser are important in decision-making process.

4) ***Execute:*** this activity is responsible for performing the decisions that are made by the decision-maker. It has the admission control which is responsible for accepting/ rejecting the requests based on the resources' availability.

B. System Model and Assumptions

This section describes the adopted system model using the layered architecture. It consists of IoT-layer, edge-layer, and cloud-layer. The cloud layer is out of the paper's scope.

The IoT layer (i.e., bottom layer) consists of many IoT devices (e.g., smartphones) and connected to the upper layer (i.e., edge layer) via a 5G cellular network. The IoT devices send a set of requests $R = \{r_i \in R \mid i = 1, 2, 3, \dots, N\}$ to the edge layer where each IoT device demands only one application type and eligible for sending more than one request. Further, each request is associated with application type a_i where a set of containerised applications are available $A = \{a_x \in A \mid x = 1, 2, 3, \dots, 6\}$, RC_x, TL_x, and UP_x where these notations are defined in Table I. The specification of these requirements is made with respect to the implementation environment which will be discussed in Section IV-C.

On the other hand, the edge layer consists of a set of ENs, $EN = \{en_j \in EN \mid j = 1, \dots, 4\}$ that are located in one cluster (i.e., a group of ENs) and connected to a centralised orchestrator that is the brain of the edge layer and hosts the proposed elasticity SAS framework. Further, we assume that the ENs are homogenous virtualised environment that hosts containerised applications A in a bare-metal manner. Further, each en_j is associated with the following capabilities: CPU_j and PS_j, which are defined in Table I.

In case of a request r_i is accepted, the request maps to a container instance based on the application type a_i. Once the request is processed, the results are sent back to the IoT device associated with the DW_i that is defined in Table I.

TABLE I. NOTATIONS

Symbol	Definition
EN	A set of edge nodes at the edge layer, where $en_j \in EN$
CPU_j	# of CPU cores for each en_j
PS_j	Processing speed of en_j
A	A set of applications requested by IoT, where $a_i \in A$
R	A set of requests generated by IoT, where $r_i \in R$
$RC_{i,x}$	Required CPU of r_i
$TL_{i,x}$	Task length of r_i
$UP_{i,x}$	Uploading data size of r_i
$DW_{i,x}$	Downloading data size of r_i
y	# of containers that needs to be instantiated/terminated

C. Proposed Algorithms

There are four algorithms that are developed to support the proposed framework. These algorithms are explained next.

1) Reactive algorithm (see Figure 2): it is a threshold-based algorithm that is responsible for reactively triggering the number of utilised containers and make instantiation decision. This algorithm is used in both reactive adaptation and hybrid adaptation. Once a request is accepted (line 1), it checks the number of stand-by (i.e., up and ready) containers. If the number of stand-by containers is below the threshold, it makes an instantiation decision (line 2 and 3).

```
Input: # of stand-by containers for each App. $Cont_{SB}^{a_x}$
and Minimum # of containers $Cont_{Min}$
Output: Elasticity decision (instantiate) by $y$
0: Begin
1: For after each accepted request of $a_i$ do
2:     If ($Cont_{SB}^{a_x} < Cont_{Min}$)
3:         Instantiate by $y$ from $a_x$
4:     End if
5: End for
6: End
```

Figure 2. Reactive algorithm (Algorithm 1).

```
Input: Predicted value $P$, # of stand-by containers for each
App. $Cont_{SB}^{a_x}$, maximum number of allowed containers
$Cont_{Max}$, and # of App. $A_{len}$.
Output: Elasticity decision (instantiate/ terminate) by $y$
0: Begin
1: For each time interval do
2:     Compute $P^{a_x} \leftarrow \left\lceil \frac{P}{A_{len}} \right\rceil$
3:     If ($P^{a_x} > Cont_{Max}$)
4:         Set $P^{a_x} \leftarrow Cont_{Max}$
5:     End if
6:     For each application $a_i$ do
7:         If ($P^{a_x} == Cont_{SB}^{a_i}$)
8:             No decision
9:         Else if ($P^{a_x} < Cont_{SB}^{a_i}$)
10:            Terminate by $y \leftarrow (Cont_{SB}^{a_x} - P^{a_x})$
11:        Else
12:            Instantiate by $y \leftarrow (P^{a_x} - Cont_{SB}^{a_x})$
13:        End if
14:    End for
15:End for
16:End
```

Figure 3. Proactive algorithm (Algorithm 2).

2) Proactive algorithm (see Figure 3): it is responsible for instantiating and terminating containers proactively using the CAPF outputs and the number of stand-by containers. First, for each time interval, it computes the predicted value for each application type (line 2). Then, it compares this value for each application type with the maximum number of allowed containers in the edge (line 3). If the predicted value is greater than the maximum number of containers, it considers the maximum (line 4). Then, for each application, it calculates the required number of containers by comparing the stand-by containers with the predicted value (line 6- 13).

Based on this calculation the decision is made either no decision, instantiate, and terminate.

3) Hybrid algorithm (see Figure 4): it brings both proactive and reactive algorithms together in the same SAS. The proactive algorithm runs at equal time intervals whereas the reactive is continuously running.

4) Admission control algorithm (see Figure 5): it is responsible for accepting/rejecting the requests based on the container's availability. Once a request is received, it checks if there is any stand-by container (line 2-6). If there is a stand-by container from the same application category, the request will be accepted and executed on the targeted containerised application. Otherwise, the request will be rejected.

```
Input: # of stand-by containers for each App. $Cont_{SB}^{a_x}$
and Minimum # of containers$Cont_{Min}$ , predicted value $P$,
maximum number of allowed containers $Cont_{Max}$, and # of
App. $A_{len}$
Output: Elasticity decision (instantiate/ terminate) by $y$
0: Begin
1: While true do
2:     For each time interval do
3:         Call proactive algorithm
4:     End for
5:     Call reactive algorithm
6: End while
7: End
```

Figure 4. Hybrid algorithm (Algorithm 3).

```
Input: # of stand-by containers for each App. $Cont_{SB}^{a_i}$ and the
App. type of received request $r_{i,x}$.
Output: Accept or reject decision
0: Begin
1: While true do
2:     If ($Cont_{SB}^{a_i}$ > 0)
3:         Accept $r_{i,x}$
4:     Else
5:         Reject $r_{i,x}$
6:     End if
7: End while
8: End
```

Figure 5. Admission control algorithm (Algorithm 4).

D. Applications Profiling

This section describes the selection of the adopted applications. In fact, selecting the application is a critical decision that must be taken carefully as it plays a major role in any resource management research. Therefore, we select a set of applications that are different in terms of *latency requirements* where the EC paradigm is mainly emerged to support latency-sensitive applications and the *resource requirements* where the EC has limited resources by nature. In other words, both latency- and resource-requirement are important concepts for EC environments as it emerges to support latency-sensitive applications as well as often has limited resources which represents a bottleneck for IoT applications. For these reasons, we classify the applications into latency-sensitive, medium-latency, and latency-tolerant applications. Similarly, the application requirements are classified into high, medium, and low requirements. Thus, the requirements are specified using ranges as these requirements do not exist in the literature with consideration to the implementation environment. For example, the required number of CPU cores ranges between 1-4 cores. Similarly, the required processing speed and servers' utilisation ranges between 500-2000 MIPS and 5%-20%, respectively. In case of the uploading and downloading data size, the assumptions are made according to the scenarios below as limited information is available in the literature. In short, we select 6 different applications, two applications from each category. This variety of both applications and their requirements is critical as it shows the effectiveness of the proposed framework and represents a realistic scenario. The main configuration parameters of adopted applications are shown in Table II. They are set to be as realistic as possible based on the real scenarios below:

TABLE II. APPLICATIONS CONFIGURATIONS

App.	CPU cores	Avg. processing speed (MIPS)	Server's utilisation (%)	Avg. upload size	Avg. download size (KB)
FR	4	2000	20	450 MB	5
ETM	4	2000	20	200 MB	5
AR	2	1000	10	1 MB	50
HM	2	1000	10	200 KB	5
IHM	1	500	5	200 KB	20
IP	1	500	5	4 KB	4

1) Face Recognition (FR): Public safety domain includes a wide range of applications, such as FR and Cars' plates identification [32][33][34]. This kind of applications is important as it helps the authority to track people, find a missing person, and track cars. It can be implemented by consuming video surveillance resources. In the context of this paper, we assume that FR application is used to find or track a person in any incident, thus, it is considered as a latency-sensitive application in the sense that the video frames need to be analysed quickly. Further, due to the type of data (video frames), the FR is categorised as a data-intensive and computational-intensive application [32][35][36].

2) Emergency Traffic Management (ETM): Nowadays, traffic flow management systems are important applications that help improving traffic efficiency, reduce accidents, support emergency services, and manage traffic jams [37]. In this paper, ETM application focuses on supporting emergency services where the application can be requested by emergency services (e.g., ambulance and fire trucks) to perform better traffic management. The ETM can be classified as a latency-sensitive application as it is related to emergency cases. It is also considered data-intensive and computational-intensive in some use cases [38]. However, in our scenario, we assume that the data size is medium as it may require gathering data from different sources (e.g., infrastructure, vehicles, and pedestrians). At the same time, ETM is computational-intensive. Based on this scenario, the application configuration is set as shown in Table II.

3) Augmented Reality (AR): AR is a well-known application that can be used in many domains, such as healthcare, agriculture, and tourism [40][41][42][43]. For the purpose of this research, we assume that it is used to guide the people during their mobility in a city by displaying contextual information about the objects/ directions by analysing the captured figures (i.e., frames). We also assume that it requires medium latency as it is not an urgent application when compared to TFM and FR. It also requires a medium-requirments of resources.

4) Health Monitoring (HM): It is a common application that benefits from EC. In general, it is classified as data-sensitive and latency-sensitive application [41][44][45][46]. However, we assume that the real-time and emergency data analysis is performed locally on the patients' smartphones or wearable devices whereas the edge layer is utilised to perform a future health prediction in a form of requests. Then, the alarm will be sent to the hospital in case of any incident is predicted. This assumption makes this application requires medium latency and resources.

5) Industrial Health Monitoring (IHM): IHM application covers a wide number of scenarios. In this research, we adopted the same scenario that is presented in [3], where the data is collected and sent to the edge for analysis and visualisation. In other words, the application is used to monitor the workers' health and environment. Further, IHM is latency-tolerant and low-requirements application.

6) Intelligent Parking (IP): IP is a smart city application which is used to search for parking slots [37]. It is assumed that the driver submitted a search request to the edge layer which is responsible for retrieving the relevant data about parking slots and suggestions to the user as a list of nearest parking spaces. Based on this scenario, IP is considered as a latency-tolerant and low-requirements application.

IV. EXPERIMENTAL DESIGN

This section presents the experimental design, which includes implementation scenarios, dataset, workload, evaluation metrics, and hypothesis.

A. *Implementation Scenarios*

The proposed elasticity SAS is implemented considering the proactive, reactive, and hybrid adaptation, separately. For instance, in the implementation of proactive adaptation, the reactive adaptation components are deactivated.

To evaluate the effectiveness of these approaches two application-based scenarios are considered as summarised in Table III. The experiments are conducted considering *1) mixed applications*, *2) single application*. Further, in each scenario, the adaptation approaches are evaluated using three real workloads which will be discussed in the next section. In the single application scenario, the adaptation approaches are evaluated considering one application in each experiment. This means each adaptation approach is evaluated using FR application as a heavy load, AR application as a medium load, and IHM application as a low load. The consideration of these scenarios is important to design the most suitable SAS for EC environment considering the adaptation approaches, workload, and applications.

TABLE III. IMPLEMENTATION SCENARIOS

Scenarios		Adaptation approaches	Considered applications
1: Mixed applications		Proactive	All
		Reactive	All
		Hybrid	All
2: Single application	A	Proactive	FR
		Reactive	FR
		Hybrid	FR
	B	Proactive	AR
		Reactive	AR
		Hybrid	AR
	C	Proactive	IHM
		Reactive	IHM
		Hybrid	IHM

B. *Workload*

This paper adopts the Shanghai Telecom dataset [46] which is ideal for the consideration of IoT workload as previously used by [48][49][50][51]. It provides six months of mobile phones records accessing the Internet and connecting to base stations that are distributed over Shanghai city.

The same workload pattern will be used as in our previous work [13]. In [13], the workload is divided into three patterns, which are decreasing, increasing, and fluctuating. Each pattern consists of a set of hours as each pattern represents a part of the day (i.e., late night and early morning as a decreasing pattern, morning as an increasing pattern, and afternoon to evening as a fluctuating pattern). Further, one hour from each pattern is selected (2nd hour from decreasing, 12th hour from increasing, and 14th hour from fluctuating) to be used in training/ testing the proposed forecasting models.

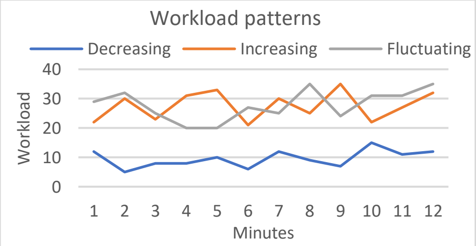


Figure 6. Workload patterns.

The testing part is used to evaluate the proposed SAS. It represents the last 12 minutes from each pattern named as decreasing, increasing, and fluctuating workload, which will be fed to the simulation environment. The workload is shown in Figure 6 over time interval. The use of 12 minutes is considered long enough to evaluate the SAS including the adaptation approaches and algorithms. Note that the workload in Figure 6 does not show the overall pattern (i.e., decreasing, increasing, or fluctuating) as it is a snip from the dataset by zooming in towards the adopted time frame by this paper. The patterns are named according to the previous work [13] to ensure consistency.

C. *Simulation Setup*

The proposed elasticity SAS framework is implemented using the EdgeCloudSim simulator [51], which is built upon the CloudSim simulator. It allows simulating the EC environment with consideration to the IoT-, edge-, and cloud-layers. It also can simulate different scenarios with/without cloud consideration and edge orchestrator. In this paper, two layers only are considered, the IoT- and edge-layers (the cloud layer out of our scope). The simulation duration is 14 minutes; the 1st minute is considered as a warm-up period and the last minute is waiting time to allow all tasks to be completed. The 12 minutes in-between is the real workload that is fed to the simulator. Further, six applications are considered with different requirements to evaluate the proposed SAS. Also, considering four ENs is deemed sufficient to allow performing the evaluation process. In terms of the number of IoT devices and requests, these values are specified for each workload pattern according to the number of devices and requests in the dataset. The most important simulation parameters are shown in Table IV.

TABLE IV. SIMULATION CONFIGURATION

Parameter	Value
Simulation time (min.)	14
Warm-up period (min.)	1
# of iterations	5
IoT Applications (mixed/ Single)	(6/1)
# of IoT devices (decreasing/ increasing/ fluctuating)	(108/277/271)
# of IoT requests (decreasing/ increasing/ fluctuating)	(115/331/334)
# of edge nodes	4
# of cores/edge node	4
Processing speed/edge node (MIPS)	2000
Resource check interval (sec.)	15

D. *Evaluation Metrics*

Two evaluation metrics are used. They are the *acceptance rate* and *servers' utilisation*. The acceptance rate evaluates the effectiveness of each adaptation approach when dealing with dynamic workload. On the other hand, the servers' utilisation refers to the CPU utilisation over the time intervals.

E. *Hypothesis*

Two hypotheses are considered to evaluate the effectiveness of adaptation approaches in the proposed elasticity SAS framework:

Hypothesis 1: *The use of the hybrid adaptation in an elasticity framework will provide the highest acceptance rate as compared to both proactive and reactive adaptations.*

Hypothesis 2: *The proactive adaptation will perform better than the reactive adaptation due to the prediction ability that helps acting prior (i.e., in advance) events happen.*

V. PERFORMANCE EVALUATION

This section evaluates and discusses the results as well as highlights the main findings.

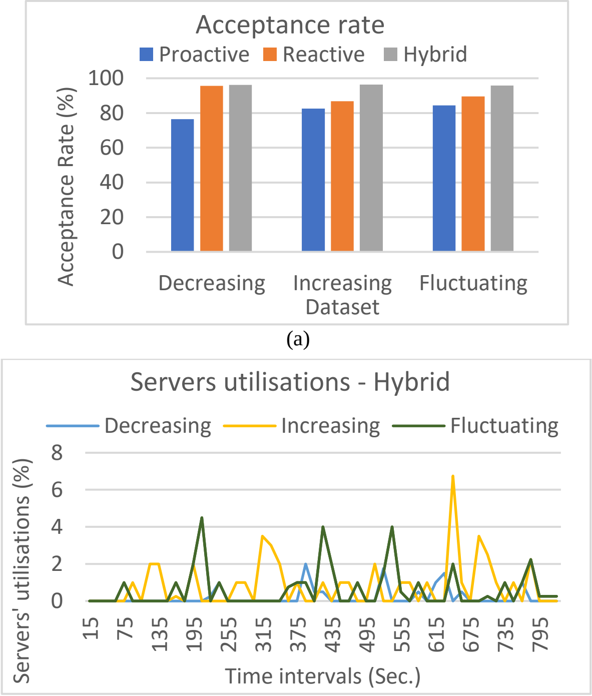
A. *Adaptation Approaches Evaluation*

The adaptation approaches are compared with respect to the stated scenarios in Section IV-A.

Scenario 1- mixed applications: it evaluates the adaptation approaches using all applications over decreasing, increasing, and fluctuating patterns.

Scenario 2- single application: it evaluates the adaptation approaches with respect to the application category (i.e., heavy-, medium-, and low-load).

1) Scenario 1 (All Apps.): the hybrid adaptation provides the highest acceptance rate overall pattern when compared to other adaptation approaches as shown in Figure 7.a. It performs about 10% higher than reactive adaptation for the increasing pattern and 7% for fluctuating pattern. It also performs about 20%, 14%, and 12% higher than proactive adaptation in decreasing, increasing, and fluctuating patterns, respectively. The hybrid adaptation superiority is due to its ability to trigger unpredicted requests thanks to the reactive adaptation side. The high acceptance rate leads to efficient utilisation of the ENs, see Figure 7.b, which shows the servers' utilisations over time for the hybrid adaptation as it has the highest acceptance rate and utilisation.

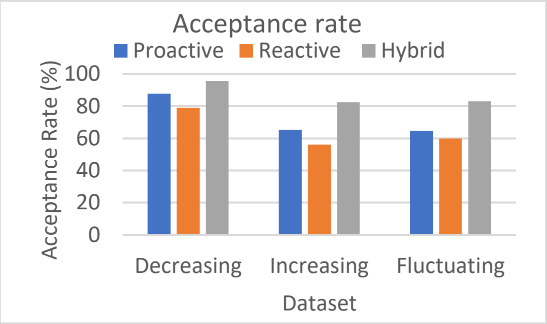


(a)

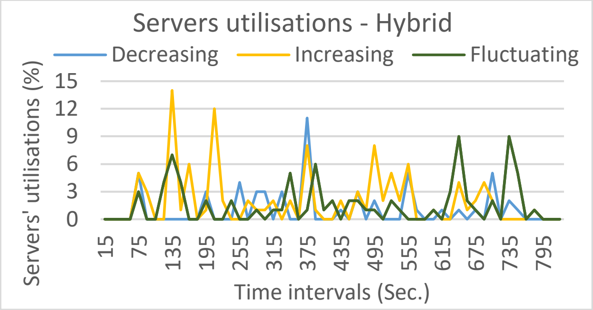
(b)

Figure 7. Scenario 1.

2) Scenario 2.A (FR): in this scenario, the hybrid adaptation also outperforms both proactive and reactive adaptation overall patterns as shown in Figure 8.a thanks to the consideration of both proactive and reactive adaptations where the reactive adaptation side can deal with unpredicted events. However, the proactive adaptation outperforms the reactive adaptation overall patterns. This due to the use of single application scenario. This means that all the submitted requests will be from the same type of application whereas in scenario 1 the predicted value will be divided over the number of considered applications assuming that all these applications come on the same probability. In terms of servers' utilisation, the FR application is considered as a heavy-load application. This means a higher workload is expected as compared to Scenario 1 which considers all applications. Figure 8.b shows the servers' utilisation over thensimulation time for the hybrid adpatation which can reach about 15% in this scenario as a heavy-load application is considered.



(a)



(b)

Figure 8. Scenario 2A.

3) Scenario 2.B (AR): in this scenario, the results are similar to Scenario 2.A where the hybrid adaptation outperforms all the adaptation approaches over different patterns as shown in Figure 9. The proactive adaptation also outperforms the reactive adaptation. The main difference is the average server utilisation which is lower than the avergae servers utilisation in the FR scenario where the AR application is considered as medium-load.

4) Scenario 2.C (IHM): the acceptance rate of this scenario is similar to Scenario 2.B which is not presented due to space limitation. In terms of the servers' utilisation, it is the lowest as compared to all previous experiments in the sense that it considers applications with low-load.

B. Hypothesis Evaluation

This section tests the hypotheses based on the considered scenarios.

1) ***Hypothesis 1:*** *The use of the hybrid adaptation in an elasticity framework will provide the highest acceptance rate as compared to both proactive and reactive adaptations.* This holds true in all scenarios. The hybrid adaptation shows a great performance as compared to both proactive- and reactive-adaptation. This is due to the consideration of the proactive adaptation to prepare the containers prior receiving the requests as well as the use of threshold-based in the reactive adaptation to maintain the number of stand-by containers.

2) ***Hypothesis 2:*** *The proactive adaptation will perform better than reactive adaptation due to prediction ability that helps acting prior (i.e., in advance) events happen.* This hypothesis is disproved for the mixed scenario (i.e., Scenario 1), while correct for the single scenarios (i.e., Scenarios 2A, 2B, and 2C). In terms of the mixed scenario, the predicted workload by the CAPF is divided by the number of applications and assuming that all applications have the same arrival probability. This means the CAPF predicts the overall workload without any consideration to the applications' arrival probability. This assumption is made as there is no previous information available in the real dataset about the type of applications that will be requested. In contrast, the proactive adaption outperforms the reactive in single scenarios as the predicted workload will be utilised by the same application.

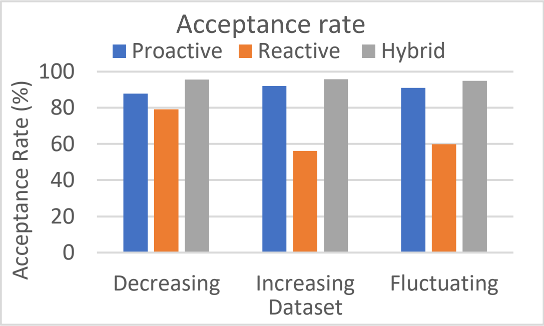


Figure 9. Scenario 2B.

C. Findings and Recommendations

The main findings of this paper can be summarised as follows with some recommendations:

1) Although the hybrid adaptation is complex and requires bringing both proactive and reactive adaptation together in a consistent manner, it provides the best performance over different scenarios and workload patterns and has the ability to adapt in a highly fluctuating environment. According to this finding, the hybrid SAS is recommended to be used in highly fluctuating environments, such as EC, as it provides a full monitoring loop with the ability to deal with unpredicted events. In other words, using either proactive or reactive adaptation approaches in a highly fluctuating environment may lead to low performance as there is a need to anticipate the future behaviour as well as using reactive adaptation for backup. The use of hybrid adaptation is also important even when the prediction models show great accuracy.

2) The hybrid adaptation efficiently utilises the EC resources as it is able to accept more requests and contributes to avoid over/under-prevision cases thanks to the use of the reactive adaptation as a back-up for the proactive adaptation to deal with the unpredicted workload. Its efficiency can be seen clearly in Scenario 1 when there is no previous knowledge about the request types. Thus, the hybrid adaptation is recommended in EC environments as these have limited resources by nature.

3) The available information about submitted requests to the edge layer plays an important role in designing the elasticity SAS framework. In fact, the proactive adaptation is preferable as compared to the reactive adaptation as it acts prior the event occurrence and prepares the resources in advance. However, in case limited information about the predicted events is available, this may lead to low performance as compared to the reactive adaptation as well as unpredicted results. This can be seen in scenario 1 when using mixed applications where the predicted value represents the overall workload without the consideration of the arrival probability for each application type.

4) It is important to evaluate the performance of the adaptation approaches in the implementation domain as their performance may vary according to the scenario and workload.

VI. CONCLUSION AND FUTURE WORK

This paper has presented and evaluated an elasticity SAS which is supported by proactive, reactive, hybrid, and admission control approaches as well as various application scenarios. The experiment results show that the most appropriate adaptation approach in an EC environment is the hybrid where its performance is at least 10% better than other approaches. The results also reveal that the performance of the adaptation approaches is domain, application, and scenario dependent.

As future work, the proposed SAS will be evaluated using different workloads aiming to stress the SAS with higher request rates. In fact, the use of higher request rate to evaluate the proposed framework is important as some experiments show small servers utilization. Additionally, both the scalability and QoS will be considered with the aim to maximise the number of running applications with adequate QoS. Moreover, a policy management will be investigated to identify the trade-off between the service acceptance maximisation from the perspective of the service provider and the QoS from the consumer, respectively.

REFERENCES

[1] P. Parwekar, “From Internet of Things towards cloud of things”, The 2nd International Conference on Computer and Communication Technology, 2011, pp. 329–333.

[2] I. Yaqoob et al., “Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges”, IEEE Wirel. Commun., vol. 24, no. 3, pp. 10–16, 2017.

[3] F. Wu, T. Wu, and M. R. Yuce, “An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications”, Sensors (Switzerland), vol. 19, no. 1, pp. 1–21 2018.

[4] J. Cheng, W. Chen, F. Tao, and C. L. Lin, “Industrial IoT in 5G environment towards smart manufacturing”, Journal of Industrial Information Integration, vol. 10 , pp. 10–19, 2018.

[5] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, “Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review”, Journal Systems and Software, vol. 136, pp. 19–38, 2018.

[6] G. Li, J. Song, J. Wu, and J. Wang, “Method of Resource Estimation Based on QoS in Edge Computing”, Wirel. Commun. and Mob. Comput., vol. 2018, pp. 1–9 ,2018.

[7] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, “Resource provisioning for IoT services in the fog computing environment: An autonomic approach”, Comput. Commun., vol. 161, pp. 109–131, 2020.

[8] B. Liu, J. Guo, C. Li, and Y. Luo, “Workload forecasting based elastic resource management in edge cloud”, Comput. & Ind. Eng., vol. 139, pp. 1–12, 2020.

[9] R. Kavanagh, K. Djemame, J. Ejarque, R. M. Badia, and D. Garcia-Perez, “Energy-aware Self-Adaptation for Application Execution on Heterogeneous Parallel Architectures”, IEEE Trans. Sustain. Comput., vol. 5, no. 1, pp. 81–94, 2019.

[10] M. D’Angelo, “Decentralized Self-Adaptive Computing at the Edge”, IEEE/ACM The 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2018, pp. 144–148.

[11] M. Xu and R. Buyya, “Brownout Approach for Adaptive Management of Resources and Applications in Cloud Computing Systems: A Taxonomy and Future Directions”, ACM Comput. Surv., vol. 52, no. 1, pp. 1–27, 2019.

[12] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, “A survey on engineering approaches for self-adaptive systems”, Pervasive Mob. Comput., vol. 17, pp. 184–206, 2015.

[13] A. F. Aljulayfi and K. Djemame, “A Machine Learning Based Context-aware Prediction Framework for Edge Computing Environments”, The 11th International Conference on Cloud Computing and Services Science, 2021, pp. 143–150.

[14] G. Galante and L. C. E. de Bona, “A Survey on Cloud Computing Elasticity”, IEEE/ACM The Fifth International Conference on Utility and Cloud Computing, 2012, pp. 263–270.

[15] M. Amiri and L. Mohammad-Khanli, “Survey on prediction models of applications for resources provisioning in cloud”, Jou. Netw. Comput. Appl., vol. 82, pp. 93–113, 2017.

[16] R. Moreno-vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, “Efficient resource provisioning for elastic Cloud services based on machine learning techniques”, Jou. Cloud Comput. Adv. Syst. and Appl., vol. 8, no. 1, pp. 1–18, 2019.

[17] D. F. Kirchoff, M. Xavier, J. Mastella, and C. A. F. De Rose, “A preliminary study of machine learning workload prediction techniques for cloud applications”, The 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2019, pp. 222–227.

[18] S. Ajila and A. A. Bankole, “Cloud Client Prediction Models Using Machine Learning Techniques”, in International Computer Software and Applications Conference, 2013, pp. 134–142.

[19] D. Spatharakis et al., “A scalable Edge Computing architecture enabling smart offloading for Location Based Services”, Pervasive Mob. Comput., vol. 67, pp. 1–15, 2020.

[20] M. Aldossary, I. Alzamil, and K. Djemame, “Towards Virtual Machine Energy-Aware Cost Prediction in Clouds”, in GECON: International Conference on the Economics of Grids, Clouds, Systems, and Services, 2017, pp. 285–299.

[21] M. Aldossary and K. Djemame, “Performance and Energy-based Cost Prediction of Virtual Machines Auto-scaling in Clouds”, The 44th Euromicro Conference on Software Engineering and Advanced Applications, 2018, pp. 502–509.

[22] M. Abdullah, W. Iqbal, A. Mahmood, F. Bukhari, and A. Erradi, “Predictive Autoscaling of Microservices Hosted in Fog Microdata Center”, IEEE Syst. Jou., pp. 1–12, 2020.

[23] A. V. Chandak and N. K. Ray, “Multi Agent Based Resource Provisioning in Fog Computing”, Trends Comput. Intell. Secur. Internet Things. Commun. Comput. Inf. Sci., vol. 1358, pp. 317–327, 2020.

[24] K. Kaur, T. Dhand, N. Kumar, and S. Zeadally, “Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers”, IEEE Wirel. Commun., vol. 24, no. 3, pp. 48–56, 2017.

[25] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, “Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks”, The 15th International Symposium on Wireless Communication Systems, 2018, pp. 1–5.

[26] S. Mireslami, L. Rakai, M. Wang, and B. H. Far, “Dynamic Cloud Resource Allocation Considering Demand Uncertainty”, IEEE Trans. Cloud Comput., pp. 1–14, 2019.

[27] E. Ataie, R. Entezari-Maleki, S. E. Etesami, B. Egger, D. Ardagna, and A. Movaghar, “Power-aware performance analysis of self-adaptive resource management in IaaS clouds”, Futur. Gener. Comput. Syst., vol. 86, pp. 134–144, 2018.

[28] G. Anders, F. Siefert, M. Mair, and W. Reif, “Proactive Guidance for Dynamic and Cooperative Resource Allocation under Uncertainties”, in International Conference on Self-Adaptive and Self-Organizing Systems, 2014, pp. 21–30.

[29] C. Kan, “DoCloud: An Elastic Cloud Platform for Web Applications Based on Docker”, The 18th International Conference on Advanced Communication Technology, 2016, pp. 478–483.

[30] C. Li, J. Bai, Y. Ge, and Y. Luo, “Heterogeneity-aware elastic provisioning in cloud-assisted edge computing systems”, Futur. Gener. Comput. Syst., vol. 112, pp. 1106–1121, 2020.

[31] K. Djemame et al., “PaaS-IaaS Inter-Layer Adaptation in an Energy-Aware Cloud Environment”, IEEE Trans. Sustain. Comput., vol. 2, no. 2, pp. 127–139, 2017.

[32] L. Soh, J. Burke, and L. Zhang, “Supporting Augmented Reality: Looking Beyond Performance”, Proc, of The 2018 Morning Workshop on Virtual Reality and Augmented Reality Network, 2018, pp. 7–12.

[33] F. Faticanti, F. D. Pellegrini, D. Siracusa, D. Santoro, and S. Cretti, “Cutting Throughput with the Edge: App-Aware Placement in Fog Computing”, The 6th IEEE International Conference on Cyber Security and Cloud Computing, and The 5th IEEE International Conference on Edge Computing and Scalable Cloud, 2019, pp. 196–203.

[34] E. Yigitoglu, M. Mohamed, L. Liu, and H. Ludwig, “Foggy: A Framework for Continuous Automated IoT Application Deployment in Fog Computing”, Proc. IEEE The 6th International Conference on AI and Mobile Services, 2017, pp. 38–45.

[35] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments”, Softw.: Pract. and Exp., vol. 47, no. 9, pp. 1275–1296, 2017.

[36] S. Yi, C. Li, and Q. Li, “A Survey of Fog Computing: Concepts, Applications and Issues”, Proc. of The 2015 Workshop on Mobile Big Data, 2015, pp. 37–42.

[37] A. Shahzad, J. Choi, N. Xiong, Y. Kim, and M. Lee, “Centralized Connectivity for Multiwireless Edge Computing and Cellular Platform: A Smart Vehicle Parking System”, Wirel. Commun. and Mob. Comput., vol. 2018, pp.1–23, 2018.

[38] S. Kekki et al., “MEC in 5G networks”, ETSI white Paper No. 28, pp. 1–28, 2018.

[39] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, “Towards Wearable Cognitive Assistance”, in Proc. of The 12th annual international conference on Mobile systems, applications, and, services, 2014, pp. 68–81.

[40] G. White, C. Cabrera, A. Palade, and S. Clarke, “Augmented Reality in IoT”, In: Liu X. et al. (eds) Service-Oriented Computing- ICSOC 2018 Workshops, Lecture Notes in Computer Science, vol. 11434, Springer, Cham , 2019, pp. 149–160.

[41] J. Dolezal, Z. Becvar, and T. Zeman, “Performance Evaluation of Computation Offloading from Mobile Device to the Edge of Mobile Network”, in IEEE Conference on Standards for Communications and Networking, 2016, pp. 1–7.

[42] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht, “Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality”, IEEE Trans. Vis. Comput. Graph., vol. 23, no. 6, pp. 1706–1724, 2017.

[43] G. Rahman and C. W. Chuah, “Fog Computing, Applications, Security and Challenges, Review”, Int. Jou. Eng. Technol., vol. 7, no. 3, pp. 1615–1621, 2018.

[44] A. C. Baktir, A. Ozgovde, and C. Ersoy, “How Can Edge Computing Benefit from Software-Defined Networking: A Survey, Use Cases, and Future Directions”, IEEE Commun. Surv. Tutorials, vol. 19, no. 4, pp. 2359–2391, 2017.

[45] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, “Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction”, Proc. IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 356–363.

[46] Sguangwang.com, “The Telecom Dataset”, 2018. [Online]. Available from: http://sguangwang.com/TelecomDataset.html. [Retrieved: 2021.5.27].

[47] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C. Hsu, “Edge server placement in mobile edge computing”, Jou. Parallel and Distrib. Comput., vol. 127, pp. 160–168, 2019.

[48] S. Wang, Y. Zhao, L. Huang, J. Xu, and C. Hsu, “QoS prediction for service recommendations in mobile edge computing”, J. Parallel Distrib. Comput., vol. 127, pp. 134–144, 2019.

[49] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C. Hsu, “User allocation-aware edge cloud placement in mobile edge computing”, Softw.: Pract. and Exp., vol. 50, no. 5, pp. 489–502, 2019.

[50] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, “Delay-aware Microservice Coordination in Mobile Edge Computing: A Reinforcement Learning Approach”, IEEE Trans. Mob. Comput., vol.20, no. 3 pp. 939–951, 2021.

[51] C. Sonmez, A. Ozgovde, and C. Ersoy, “EdgeCloudSim: An environment for performance evaluation of edge computing systems”, Trans. Emerg. Telecommun. Technol., vol. 29, no. 11, pp. 1–17, 2018.

Centralised Autonomic Self-Adaptation in a Foraging Robot Swarm

Liam McGuigan, Roy Sterritt, George Wilkie
School of Computing, Faculty of Computing, Engineering and the Built Environment
Ulster University
Jordanstown, N. Ireland
email: mcguigan-l8@ulster.ac.uk, r.sterritt@ulster.ac.uk, fg.wilkie@ulster.ac.uk

***Abstract*— The use of robotic swarms in domains, such as space exploration, or search and rescue missions, requires that the swarms be self-adaptive in order to adjust to newly acquired data, and react to unforeseen events. Research on swarm self-adaptation tends to focus on the adaptation of individual agents, however taking a top-down approach can allow for the use of knowledge that is only apparent considering the swarm as a whole. This research makes use of a centralised Autonomic Manager to modify the behaviour of a simulated swarm of foraging robots by adjusting the range over which individual robots broadcast help requests. The swarm is able to learn its own size and the size of the test area, and use that information to guide its decision making, showing the potential for a future decentralised approach. First, the swarm is tasked with recognising the initial situation. Secondly, the swarm must respond to two events which alter the scenario parameters, namely the destruction of a proportion of the swarm, and a change in effective communication range. Performance of the swarm using an Autonomic Manager is compared against that using a fixed broadcast range suited to the initial circumstances. The results show that the swarm can recognise the initial situation and select a suitable broadcast range. It is also capable of recognising the events that occur, but the effectiveness of its response depends on additional parameters in the simulation.**

Keywords- Swarm robotics; Self-adaptation; Autonomic Computing; Simulation.

I. INTRODUCTION

A swarm of robots, in which the aggregate behaviour of many relatively simple individuals combines to create a more complex set of behaviours [1], can have applications in areas, such as mine clearance [2], search & rescue [3] and space exploration [4][5]. A robot swarm can reduce the demands on any single robot, may accomplish the task more quickly, and can be deployed where sending humans is too dangerous, difficult, or costly.

The ability to self-adapt, that is to adjust behaviour in response to newly acquired information without the need for external guidance, is a requirement of a robotic swarm [6]. Unforeseen events may occur that require adjustment, and factors, such as distance and time, may restrict the ability for a human operator to act successfully. Self-adaptation can be applied to the swarm in a variety of ways [7], including the development of emergent behaviours [8], evolutionary systems [9] and swarm-level decision making [10].

Autonomic Computing concepts [11][12] can be used for swarm-self adaptation. At the swarm level, an Autonomic Manager (AM) employing a control loop, such as the Monitor, Analyse, Plan, Execute system described by [11] can be used to allow the swarm to assess the current situation and take any action necessary. This may be implemented in a centralised manner, with individual robots communicating with a central command unit, or decentralised with each robot using its own control loop in order to modify its own behaviour in response to shared information and experience.

The objective of this work is to explore the potential for using swarm-level self-adaptation in a swarm of robots to improve performance in a foraging task, specifically the time it takes the swarm to complete the task which may often be an application priority, such as in search & rescue.

Robot swarms are typically decentralised in nature [2], however here a centralized approach is used as an initial exploratory stage to determine if an AM provides any benefits, with the work of decentralisation to follow this research. As such, the centralised AM here is limited to analysing aggregate data and adjusting parameters, rather than taking a more active role in coordinating the swarm.

The AM aims to achieve performance improvement through modification of the range at which individual robots communicate with neighbouring robots for assistance. The swarm is tasked with deciding the appropriate communication range, and then two unforeseen events are introduced. The first, robot destruction, tests the swarm’s ability to react to the sudden change in swarm size, such as a loss of robots in a search & rescue task due to the hazardous environment. The second, a change to communications quality, represents a situation where the ability of the robots to communicate with each other may be hampered by a change in environmental conditions.

The rest of this paper is structured as follows. Section II discusses related work in swarm-level adaptation. Section III describes the simulation used and swarm task, the implementation of autonomic behaviour in the swarm, and the scenarios tested. Section IV reports these results and explores the implications. Section V concludes the paper with a summary, and future research directions.

II. RELATED WORK

The location where adaptation is applied to a swarm is important when considering the intended goal. Much of the research in adaptation focuses on the level of the individual agent, where the resulting swarm performance is affected by the aggregate of these individual behaviours [7]. This level of adaptation can have a dramatic impact on performance, but it is difficult for any single robot to take advantage of information that is only available when viewing the larger picture, or to make decisions affecting the behaviour of other members of the swarm, such as cooperation or communication.

Adaptation at the swarm level can counter some of these problems. [13] describes an approach to moderating the size of the swarm in order to reduce degraded performance due to congestion. Robots keep track of the conflicts that occur when two robots attempt to occupy the same cell. If the number of conflicts crosses a threshold, virtual pheromones can be deposited at the entrance in order to instruct robots to leave or join the area. Hence, the swarm can adjust its size based on the combination of each robot's collision tracking data.

In [14], a group of unmanned aerial vehicles (UAVs) are patrolling an area defined by a set of cells, with the aim of ensuring that cells are visited often enough during the mission. Individual UAVs decide their next target on the basis of values assigned to the cells by a central system based on UAV visitation. Different strategies for assigning those values are explored, and so the central system becomes an effective behaviour adaptation method for the group.

As discussed in Section I, autonomic concepts may be used for swarm self-adaptation. [15] describes an adaptation pattern in which one robot in the swarm takes on the role of an AM, running a control loop with visibility of the whole system. In the case study presented, the swarm was tasked with exploring an unknown area. Robots communicate their positional and explorational information with the AM, which can direct them to underexplored areas. Recognising that a centralised system may be a bottleneck, a decentralised variant is also used in which the robots share the information with their neighbours. Both approaches perform much better than a basic pheromone-based approach.

A partially distributed approach described in [16] uses a group of UAVs, together with communication base stations taking on the role of AMs, engaged in a search task. If one of the UAVs leaves the active area and loses the communication link, the base stations are able to recognise the failure and reposition themselves in order to retrieve the UAV, while also minimising disruption to the rest of the swarm.

In a previous paper [7], cooperation strategies for swarms were investigated to determine the potential for using an AM to select between them based on the situation. This research builds on that by using a centralised AM is employed to modify the broadcast range parameter to explore how an AM can improve performance over using a fixed strategy.

III. SIMULATION SETUP

The following subsections introduce the simulation setup and describe the specific task the swarm must carry out. This is followed by details of how the autonomic management of the swarm functions, and a description of the test scenarios run.

A. *Simulation and Task Description*

This research employs a time-stepped simulation of a heterogeneous swarm of agents engaged in a variant of a foraging task, as reported in previous work [7].

The simulation creates a world with a rectangular grid of cells, seeded with several items with an associated type, and several robots with corresponding types, as shown in Figure 1. Only one item may be generated on a single cell, however any number of robots may stack. A cell can be considered to represent a much larger area than the footprint of a single robot, leaving plenty of room for multiple robots per cell, thus allowing the simulation to ignore potential collisions. The simulation proceeds in a time-stepped manner – each tick of the simulation, all robots are updated in turn.

Each map is seeded with several items, which have an associated type, and several robots with corresponding types. Robots will initially search for items using a wander behaviour, selecting a random, valid direction each update to move, and moving one cell in that direction. On finding an item, the robot will forage if it matches the type, however if the types differ the robots may broadcast a help message to recruit a suitable robot within range. Foraging is carried out in-situ, rather than returning an item to a base. The process may be considered analogous to applications, such as mine deactivation, analysis of mineral deposits, or environmental clean-up.

The cooperation process is the Help Recruitment strategy as described in the previous work [7] – the robot broadcasts the help message and waits for responses. If multiple responses are received, the nearest robot is selected and assigned the task. While a robot waits for responses or task assignment, it remains stationary until the process is complete.

Communication messages are queued and processed at the end of each simulation tick. First, each message is sent to all robots within range. After all messages are sent, each robot shuffles the list of unread messages, and then processes each in turn – in this way, the simulation can avoid the update order being a factor in the behaviour of the robots. Without shuffling, if a robot was to receive two help requests in a single tick, it would always respond only to the first one.

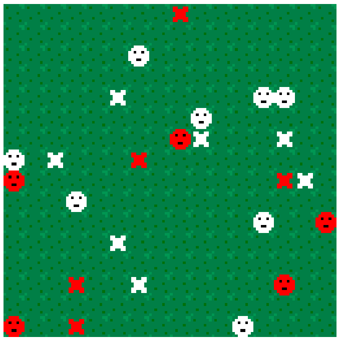


Figure 1. A portion of the world state during a simulation. The colour of a robot (face) or item (cross) indicates its type.

Communications are affected by a global quality setting, which acts as a multiplier on the range of each message sent – a value of 50%, for example, results in a message intended to be sent 10 cells to only reach robots within 5 cells.

A measure of energy expended by each robot is tracked by assigning actions an energy cost. Each robot incurs an upkeep cost of one unit per tick, in addition to the cost of actions taken. Foraging an item costs 1 unit, and movement costs 1 unit per cell moved, and 1.41 units for diagonal movement. Communication cost depends on the maximum range of the broadcast according to the power law stated in (1), where r is the range of the broadcast in cells.

$$\text{cost} = 0.01 \times r^2 \tag{1}$$

Energy is measured in arbitrary units and is designed as a means of exploring the potential impact on a swarm of using increased broadcast ranges.

B. *Autonomic Robots*

Each robot contains an autonomic management component to gather and process information local to the robot, which is then sent to a centralised AM to make swarm-wide decisions, which occurs every 32 simulation ticks. This value was chosen to balance the need to react to situations with the desire to avoid an increase in communications needed to allow for higher AM update rates.

Each robot keeps track of the rectangular region of the map it has so far explored, and sends a synchronisation message to the central AM containing that, and the robot's type. The central AM uses the aggregate data of all robots to estimate the total map size as a rectangle containing all individually explored regions, as well as the swarm size and composition by totalling individual robot types.

Additionally, each robot sends a pulse message with the same period as the central AM message. This pulse is sent to other robots within a fixed range of 8 cells to allow identification of neighbours. Before sending the message to the central AM, the robot calculates the maximum distance from which it received a pulse message from other robots, and sends this to the central AM, which in turn records the maximum distance received by any robot. This can be used together with the known pulse range of 8 cells to detect any possible changes in communications quality.

The central AM uses the information received to determine the best range at which to broadcast help requests, seeking to balance the need for a broadcast to reach a recipient, with the increased energy requirements of broadcasting at higher ranges and the impact on the swarm performance as more robots receive and respond to help requests.

To do this, the swarm uses the map size and swarm composition to calculate the density, δ, of the robots of each type within the world, as in (2),

$$\delta = r / a, \tag{2}$$

where r is the number of robots of a given type, and a is the total area of the map.

The ideal broadcast range was determined by measuring the number of simulation ticks it takes the swarm to complete the task under a selection of broadcast ranges and swarm sizes, and selecting the ranges with the shortest ticks to completion for each size, as shown in Figure 2. Fitting an approximate trend line to the plot leads to an equation for determining the broadcast range based on the lowest density of any given robot type, as in (3).

$$\text{range} = 2.6594 \times \delta_{\min}^{-0.46} \tag{3}$$

The ideal range can then be divided by the estimated communications quality in order to determine a suitable range to counter its effects. Finally, the range used is clamped between 1 and 128 before being communicated to the individual robots.

If necessary, the central AM can also decide to halt any attempt at cooperation. If communications quality drops to zero, there is no need to spend time sending messages and waiting for replies when those messages will never arrive.

C. *Test Scenarios*

Three sets of tests were conducted. First, the central AM performance was measured in set of fixed scenarios. Second, the ability of the AM to react to the sudden destruction of a proportion of the robots was tested. The third test tested the AM's ability to react to a change in communications quality.

Each test was carried out with a 128x128 map, seeded with 256 items, equally distributed between two types. Each scenario within a test was run 100 times to obtain a sample, and performance has been measured based on simulation ticks to complete the task. Additionally, the energy cost during the task has been measured.

1) *Central AM Performance:*

To test the hypothesis that the AM is capable of selecting a suitable broadcast range and perform no worse than the best fixed setting, three sets of scenarios were used, consisting of 64, 128 and finally 256 robots, equally distributed between the two item types. Each scenario was run with a set of 9 fixed help recruitment broadcast ranges set at 4, 8, 16, 24, 32, 40, 48, 56 and 64

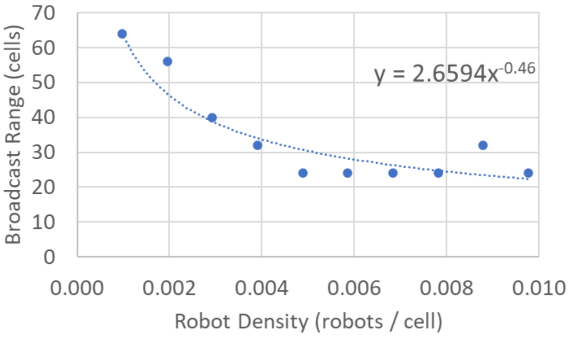


Figure 2. Derivation of the ideal broadcast range function. The points indicate the best performing broadcast range tested for the given density, based on mean simulation ticks to completion.

cells. Finally, each scenario was run with an active autonomic management system to select the best broadcast range.

2) *Robot Destruction:* To test the central AM's ability to recognise a sudden change in swarm composition, the 256-robot scenario was run with an event scheduled to occur after 300 simulation ticks, in which a given percentage of robots – equally split between the two types – is destroyed. The percentages employed in the test were 25%, 50%, 75% and 90%. Tests were run with the best performing fixed help broadcast range, as identified during the Central AM Performance test above, and then again with the active AM.

3) *Communications Quality Change:* To test the central AM's ability to recognise a change in the communications quality, the 256-robot scenario was run with an event scheduled to occur after 300 simulation ticks, in which the communication quality changes. The changes employed were 100-25%, 25-100%, 100-0% and 0-100%. As before, tests were run with the best performing fixed broadcast range, and then again with the AM.

IV. RESULTS

The following subsections discuss the results of the three main test scenarios, followed by an overall summary.

A. Central AM Performance

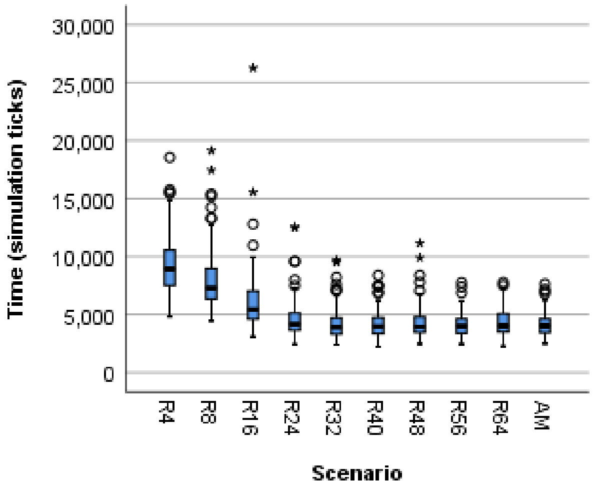
Figure 3 shows the performance of the swarm in each strategy for the three swarm sizes tested, while Figure 4 shows the total energy cost during the test for a swarm of 256 robots.

Independent t-tests were performed between the identified best broadcast range for each swarm size, against the performance of the central AM. The results of this are summarised in Table I.

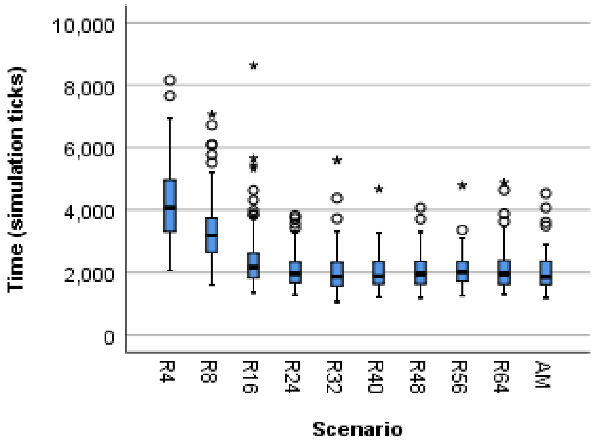
The results show that a swarm using a central AM, making decisions based on the aggregate data collected by individual robots, is capable of selecting an appropriate broadcast range for the Help Recruitment strategy used. It can be seen from the t-test results that there is no statistical difference between the best performing fixed range, and the use of an AM, at a 95% confidence level. This applies for all three swarm sizes tested.

Figure 3 (c) and Figure 4 show that while broadcast ranges of 16 cells and higher show similar performance when measured on completion time alone, the energy demands on the swarm increase when the range grows beyond 24 cells. Therefore, it is not sufficient to set the swarm to operate with a higher broadcast range in order to cover any eventuality, as the swarm would become less efficient.

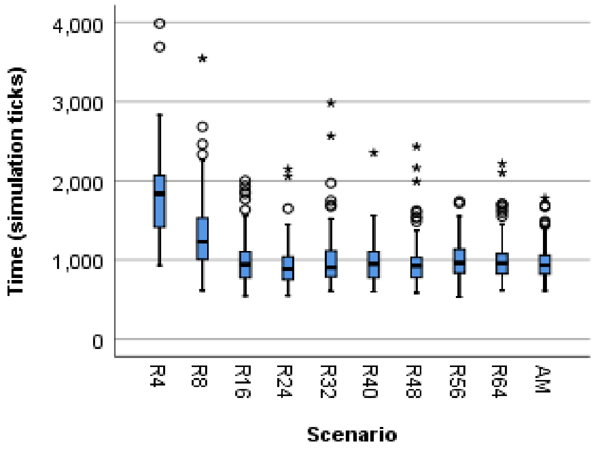
These findings show that in situations where the swarm size and operating area cannot be predicted ahead of time, the ability to determine a suitable broadcast range during the task can prove beneficial to overall swarm performance.



(a) 64 robots



(b) 128 robots



(c) 256 robots

Figure 3. Simulation ticks to complete foraging task for each broadcast range tested, and with an AM, for three different swarm sizes: (a) 64 robots, (b) 128 robots, and (c) 256 robots. Circles and crosses indicate outliers in the data.

TABLE I. CENTRAL AM PERFORMANCE T-TEST RESULTS

Swarm Size	**Ideal Range**	**Fixed Range**		**AM**		**Deg. of Freedom**	**t-statistic**	**p-value**
		Mean	Std. Dev.	Mean	Std. Dev.			
64	56	4151.59	1031.553	4153.54	1044.386	198	-0.013	0.989
128	40	2022.26	564.917	2024.05	596.051	198	-0.022	0.983
256	24	936.90	267.334	983.77	248.969	198	-1.283	0.201

TABLE II. ROBOT DESTRUCTION T-TEST RESULTS

Destroyed Robots / %	**Fixed Range**		**AM**		**Deg. of Freedom**	**t-statistic**	**p-value**
	Mean	Std. Dev.	Mean	Std. Dev.			
25	1187.51	342.602	1150.36	303.995	198	0.811	0.418
50	1630.59	617.813	1811.13	606.876	198	-2.085	0.038
75	3664.05	1709.930	3041.41	1226.075	179.516	2.959	0.004
90	11196.18	5458.347	7974.65	2753.011	146.305	5.270	0.000

B. Robot Destruction

Figure 5 shows the performance of the swarm under each test scenario. Independent t-tests were run comparing the fixed range performance with that where the AM is active, and the results are summarised in Table II.

The results here are not so clear cut. In the cases where 75% and 90% of robots are destroyed, the AM's ability to adjust the broadcast range to compensate for the decreased swarm density proves beneficial to the overall swarm performance. In situations where robots may be lost due to hazardous environments, this would prove useful.

However, the t-test results in Table II show that at the 50% level, the AM actually reduces overall performance. This result is surprising given destroying 50% of the robots leads to a remaining swarm of 128 robots, and the results of the Central AM Performance tests show that the AM performs as well as the case with a fixed broadcast range of 24 cells. Further investigation was conducted by running the fixed range and AM tests in this case a further 500 times each. The results of that test show no statistical difference between the two cases, suggesting random chance was responsible for the results in Table II for the 50% destruction test.

C. Communications Quality Change

Figure 6 shows the performance of the swarm and the energy cost for each communications quality change scenario.

Independent t-tests were run comparing the fixed range performance with that where the AM is active, and the results are summarised in Table III. The equivalent tests comparing energy usage are shown in Table IV.

The results show that the AM only improves in both performance and efficiency in the situation where the communications quality drops from 100% to 25%. The AM is able to adjust the broadcast range to compensate for the decreased communications range.

Where quality increases from 25% to 100%, the AM does not show any performance advantage. This is likely

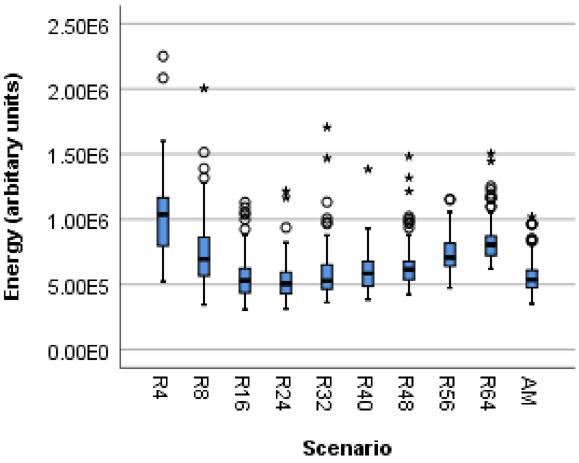


Figure 4. Energy cost for the swarm of 256 robots during the foraging task, for each broadcast range tested, and with an AM. Circles and crosses indicate outliers in the data.

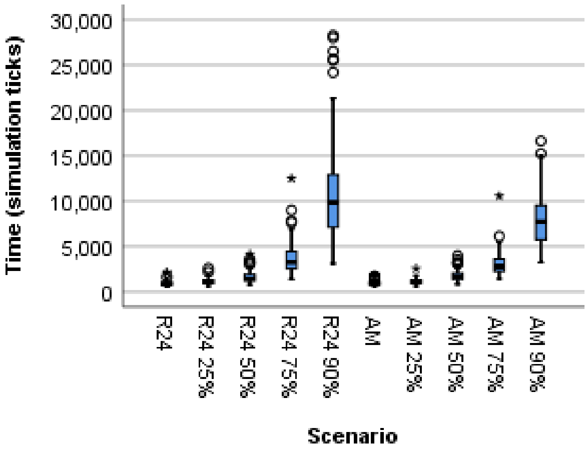


Figure 5. Simulation ticks to complete foraging task for each robot destruction scenario, with an without an AM. Circles and crosses indicate outliers in the data

due to the higher number of items during the earlier stages of the task. As help requests cause robots to stop the search for a while to participate in the recruitment process, higher ranges cause more robots to halt what would otherwise be a fruitful random search. In this scenario, the AM is also less efficient, a consequence of broadcasting at a higher range.

In the cases where the communications quality begins or ends at 0%, no statistical differences can be seen between the AM and a fixed broadcast range. This is likely because at 0% communications quality, no cooperation is possible, and the performance of the swarm is dominated by the random search for items.

D. Summary

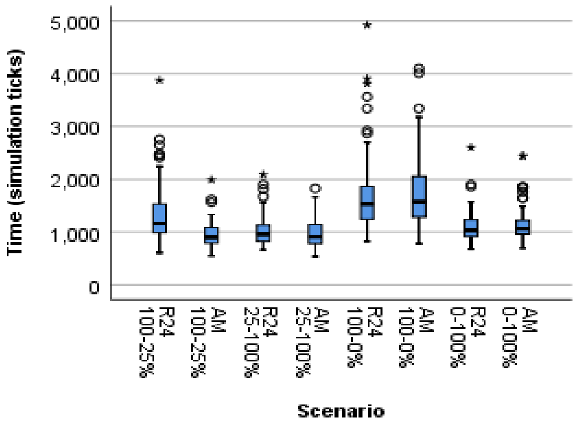
The results above show that the presence of an Autonomic Manager can have benefits for the performance of the swarm, however it is possible for the AM to reduce performance in some circumstances. These situations will require further investigation, and the AM may need to be improved in order to take into account further variables in order to counter their effects. For example, if estimates of the density of items in the world can be made, this could be used to reduce communication range when the density is high, avoiding the interruptions that may lead to the poorer performance in this period.

V. CONCLUSION AND FUTURE WORK

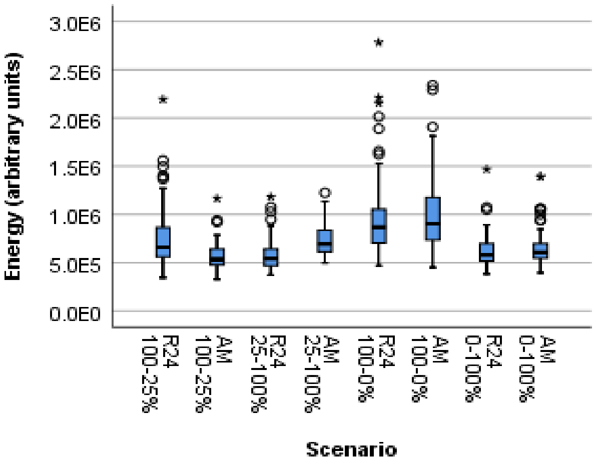
This research used a simulation of a robotic swarm equipped with a centralised Autonomic Manager capable of managing performance through the adjustment of the intra-swarm communication range.

The findings show that an AM is capable of finding an appropriate communication range when given a task where the map size and number of robots in the swarm is not initially known to the AM and must be deduced from information gathered by individual robots.

When a robot destruction event occurs, the AM proves beneficial to the swarm when the robot loss is high, capable



(a) Simulation ticks to complete



(b) Energy cost

Figure 6. Performance of the swarm in the communications quality change scenarios, with and without an AM: (a) simulation ticks to complete, and (b) energy cost. Circles and crosses indicate outliers in the data.

TABLE III. COMMUNICATIONS QUALITY CHANGE T-TEST RESULTS - TICKS

Quality Change	**Fixed Range**		**AM**		**Deg. of Freedom**	**t-statistic**	**p-value**
	Mean	Std. Dev.	Mean	Std. Dev.			
100 – 25%	1314.86	518.360	944.96	239.680	139.481	6.477	0.000
25 – 100%	1015.27	268.303	980.89	286.952	198	0.875	0.383
100 – 0%	1663.61	706.816	1724.27	625.697	198	-0.643	0.521
0 – 100%	1108.01	276.859	1132.90	302.442	198	-0.607	0.545

TABLE IV. COMMUNICATIONS QUALITY CHANGE T-TEST RESULTS - ENERGY

Quality Change	**Fixed Range / 1000**		**AM / 1000**		**Deg. of Freedom**	**t-statistic**	**p-value**
	Mean	Std. Dev.	Mean	Std. Dev.			
100 – 25%	746.49	292.797	563.98	136.668	140.184	5.648	0.000
25 – 100%	574.95	151.535	730.59	163.040	198	-6.993	0.000
100 – 0%	943.32	399.203	986.42	356.575	198	-0.805	0.422
0 – 100%	627.33	156.363	646.28	172.379	198	-0.814	0.417

of completing the task faster than using a fixed broadcast range. No benefit is seen when the robot loss is low.

In the event of a change in communications quality, the AM is capable of improving performance when the quality drops from high to low without dropping out entirely, but not when the quality starts low and increases. This is likely due to the increased item density during the early stages of the task, and it is worth exploring this factor to see how the AM might measure and take item density into account.

It is noted that this work was conducted using a centralised AM that makes global decisions on behalf of the swarm. Such a system introduces problems that have not been replicated in this work, such as the potential for the central AM to be a bottleneck on performance, the presence of a single point of failure, the need for individual robots to maintain that link, and reduced autonomy of any one robot. Future work will include producing a decentralised autonomic layer within the swarm, where individual robots run their own AMs that make decisions based on local data and swarm-level information that can be shared through the regular pulse messages.

Future work may also explore other situations that may affect performance, such as more complex maps containing obstacles, differing distributions of robots, more complexity in the foraging task, on-board batteries that drain and require recharging, and further events that may occur to unexpectedly change the world state.

REFERENCES

[1] H. Hamann, *Swarm robotics: A formal approach*. Springer International Publishing, Cham, Switzerland, 2018.

[2] I. Navarro and F. Matía, "An Introduction to Swarm Robotics", *ISRN Robot.*, vol. 2013, pp. 1–10, 2013.

[3] L. Bayindir, "A review of swarm robotics tasks", *Neurocomputing*, vol. 172, pp. 292–321, Jan. 2016.

[4] R. Sterritt, G. Wilkie, C. Saunders, M. D. C. Gama, G. Hawe, and L. McGuigan, "Inspiration for Space 2.0 from Autonomic-ANTS (Autonomous NanoTechnology Swarms) Concept missions", presented at the Reinventing Space Conference, 2019.

[5] A. Farahani, G. Cabri, and E. Nazemi, "Self-* properties in collective adaptive systems", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, Heidelberg, Germany, Sep. 2016, pp. 1309–1314.

[6] G. Beni, "From Swarm Intelligence to Swarm Robotics", in *Swarm Robotics*, Berlin, Heidelberg, 2005, pp. 1–9.

[7] L. McGuigan, C. Saunders, R. Sterritt, and G. Wilkie, "Cooperation Strategies in a Time-Stepped Simulation of Foraging Robots", in *The Twelfth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2020) IARIA*, Oct. 2020, p. 8.

[8] J. Prasetyo, G. De Masi, and E. Ferrante, "Collective decision making in dynamic environments", *Swarm Intell.*, vol. 13, no. 3, pp. 217–243, Dec. 2019.

[9] N. Bredeche, E. Haasdijk, and A. Prieto, "Embodied evolution in collective robotics: A review", *Front. Robot. AI*, vol. 5, p. 12, 2018.

[10] C. Saunders, R. Sterritt, and G. Wilkie, "Collective Communication Strategies for Space Exploration", *J. Br. Interplanet. Soc.*, vol. 72, no. 12, pp. 416–430, 2019.

[11] J. O. Kephart and D. M. Chess, "The vision of autonomic computing", *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[12] E. Vassev, R. Sterritt, C. Rouff, and M. Hinchey, "Swarm Technology at NASA: Building Resilient Systems", *IT Prof.*, vol. 14, no. 2, pp. 36–42, Mar. 2012.

[13] J. Zelenka, T. Kasanický, and I. Budinská, "A Self-adapting Method for 3D Environment Exploration Inspired by Swarm Behaviour", in *Advances in Service and Industrial Robotics*, Cham, 2018, pp. 493–502.

[14] K. S. Kappel, T. M. Cabreira, J. L. Marins, L. B. de Brisolara, and P. R. Ferreira, "Strategies for Patrolling Missions with Multiple UAVs", *J. Intell. Robot. Syst.*, vol. 99, pp. 499–515, Sep. 2019.

[15] M. Puviani, G. Cabri, and F. Zambonelli, "Agent-based Simulations of Patterns for Self-adaptive Systems":, in *Proceedings of the 6th International Conference on Agents and Artificial Intelligence*, ESEO, Angers, Loire Valley, France, 2014, pp. 190–200.

[16] M. Allison, M. Robinson, and G. Rusin, "An Autonomic Model-Driven Architecture to Support Runtime Adaptation in Swarm Behavior", in *Advances in Information and Communication*, Cham, 2020, pp. 422–437.