CLOUD COMPUTING 2022

The Thirteenth International Conference on Cloud Computing, GRIDs, and Virtualization

ISBN: 978-1-61208-948-5

April 24 - 28, 2022

Barcelona, Spain

CLOUD COMPUTING 2022 Editors

Sebastian Fischer, Technical University of Applied Sciences OTH Regensburg, Germany

CLOUD COMPUTING 2022

Forward

The Thirteenth International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2022), held on April 24 - 28, 2022, continued a series of events targeted to prospect the applications supported by the new paradigm and validate the techniques and the mechanisms. A complementary target was to identify the open issues and the challenges to fix them, especially on security, privacy, and inter- and intra-clouds protocols.

Cloud computing is a normal evolution of distributed computing combined with Service-oriented architecture, leveraging most of the GRID features and Virtualization merits. The technology foundations for cloud computing led to a new approach of reusing what was achieved in GRID computing with support from virtualization.

The conference had the following tracks:

- Cloud computing
- Computing in virtualization-based environments
- Platforms, infrastructures and applications
- Challenging features
- New Trends
- Grid networks, services and applications

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the CLOUD COMPUTING 2022 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 that dedicated much of their time and effort to contribute to CLOUD COMPUTING 2022. 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 also gratefully thank the members of the CLOUD COMPUTING 2022 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that CLOUD COMPUTING 2022 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of cloud computing, GRIDs and virtualization. We also hope that Barcelona provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city

CLOUD COMPUTING 2022 Steering Committee

Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Yong Woo Lee, University of Seoul, Korea
Bob Duncan, University of Aberdeen, UK
Alex Sim, Lawrence Berkeley National Laboratory, USA
Sören Frey, Daimler TSS GmbH, Germany
Andreas Aßmuth, Ostbayerische Technische Hochschule (OTH) Amberg-Weiden, Germany
Uwe Hohenstein, Siemens AG, Germany
Magnus Westerlund, Arcada, Finland

CLOUD COMPUTING 2022 Publicity Chair

Lorena Parra, Universitat Politecnica de Valencia, Spain
Javier Rocher, Universitat Politècnica de València, Spain

CLOUD COMPUTING 2022

Committee

CLOUD COMPUTING 2022 Steering Committee

Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Yong Woo Lee, University of Seoul, Korea
Bob Duncan, University of Aberdeen, UK
Alex Sim, Lawrence Berkeley National Laboratory, USA
Sören Frey, Daimler TSS GmbH, Germany
Andreas Aßmuth, Ostbayerische Technische Hochschule (OTH) Amberg-Weiden, Germany
Uwe Hohenstein, Siemens AG, Germany
Magnus Westerlund, Arcada, Finland

CLOUD COMPUTING 2022 Publicity Chair

Lorena Parra, Universitat Politecnica de Valencia, Spain
Javier Rocher, Universitat Politècnica de València, Spain

CLOUD COMPUTING 2022 Technical Program Committee

Omar Aaziz, Sandia National Laboratories, USA
Sherif Abdelwahed, Virginia Commonwealth University, USA
Maruf Ahmed, The University of Technology, Sydney, Australia
Abdulelah Alwabel, Prince Sattam Bin Abdulaziz University, Kingdom of Saudi Arabia
Mário Antunes, Polytechnic of Leiria, Portugal
Ali Anwar, IBM Research, USA
Filipe Araujo, University of Coimbra, Portugal
Andreas Aßmut, Ostbayerische Technische Hochschule (OTH) Amberg-Weiden, Germany
Odiljon Atabaev, Andijan Machine-Building Institute, Uzbekistan
Luis-Eduardo Bautista-Villalpando, Autonomous University of Aguascalientes, Mexico
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Mehdi Belkhiria, University of Rennes 1 | IRISA | Inria, France
Leila Ben Ayed, National School of Computer Science | University of Manouba, Tunisia
Andreas Berl, Technische Hochschule Deggendorf, Germany
Simona Bernardi, University of Zaragoza, Spain
Dixit Bhatta, University of Delaware, USA
Anirban Bhattacharjee, National Institute of Standards and Technology (NIST), USA
Peter Bloodsworth, University of Oxford, UK
Jalil Boukhobza, University of Western Brittany, France
Marco Brocanelli, Wayne State University, USA
Antonio Brogi, University of Pisa, Italy
Roberta Calegari, Alma Mater Studiorum-Università di Bologna,Italy

Paolo Campegiani, Bit4id, Italy
Juan Vicente Capella Hernández, Universitat Politècnica de València, Spain
Roberto Casadei, Alma Mater Studiorum - Università di Bologna, Italy
Ruay-Shiung Chang, National Taipei University of Business, Taipei, Taiwan
Ryan Chard, Argonne National Laboratory, USA
Batyr Charyyev, Stevens Institute of Technology, USA
Hao Che, University of Texas at Arlington, USA
Yue Cheng, George Mason University, USA
Enrique Chirivella Perez, University West of Scotland, UK
Claudio Cicconetti, National Research Council, Italy
Daniel Corujo, Universidade de Aveiro | Instituto de Telecomunicações, Portugal
Noel De Palma, University Grenoble Alpes, France
Mª del Carmen Carrión Espinosa, University of Castilla-La Mancha, Spain
Chen Ding, Ryerson University, Canada
Karim Djemame, University of Leeds, UK
Ramon dos Reis Fontes, Federal University of Rio Grande do Norte, Natal, Brazil
Bob Duncan, University of Aberdeen, UK
Steve Eager, University West of Scotland, UK
Nabil El Ioini, Free University of Bolzano, Italy
Rania Fahim El-Gazzar, Universty of South-Eastern Norway, Norway
Ibrahim El-Shekeil, Metropolitan State University, USA
Levent Ertaul, California State University, East Bay, USA
Javier Fabra, Universidad de Zaragoza, Spain
Fairouz Fakhfakh, University of Sfax, Tunisia
Hamid M. Fard, Technical University of Darmstadt, Germany
Umar Farooq, University of California, Riverside, USA
Tadeu Ferreira Oliveira, Federal Institute of Science Education and Technology of Rio Grande do Norte, Brazil
Jan Fesl, Institute of Applied Informatics - University of South Bohemia, Czech Republic
Sebastian Fischer, University of Applied Sciences OTH Regensburg, Germany
Stefano Forti, University of Pisa, Italy
Sören Frey, Daimler TSS GmbH, Germany
Somchart Fugkeaw, Sirindhorn International Institute of Technology | Thammasat University, Thailand
Katja Gilly, Miguel Hernandez University, Spain
Jing Gong, KTH, Sweden
Poonam Goyal, Birla Institute of Technology & Science, Pilani, India
Nils Gruschka, University of Oslo, Norway
Jordi Guitart, Universitat Politècnica de Catalunya - Barcelona Supercomputing Center, Spain
Saurabh Gupta, Graphic Era Deemed to be University, Dehradun, India
Seif Haridi, KTH/SICS, Sweden
Paul Harvey, Rakuten Mobile, Japan
Herodotos Herodotou, Cyprus University of Technology, Cyprus
Uwe Hohenstein, Siemens AG Munich, Germany
Soamar Homsi, Air Force Research Laboratory (AFRL), USA
Anca Daniela Ionita, University Politehnica of Bucharest, Romania
Mohammad Atiqul Islam, The University of Texas at Arlington, USA
Saba Jamalian, Roosevelt University / Braze, USA
Fuad Jamour, University of California, Riverside, USA

Weiwei Jia, New Jersey Institute of Technology, USA
Carlos Juiz, University of the Balearic Islands, Spain
Sokratis Katsikas, Norwegian University of Science and Technology, Norway
Attila Kertesz, University of Szeged, Hungary
Zaheer Khan, University of the West of England, Bristol, UK
Ioannis Konstantinou, CSLAB - NTUA, Greece
Sonal Kumari, Samsung R&D Institute, India
Van Thanh Le, Free University of Bozen-Bolzano, Italy
Yong Woo Lee, University of Seoul, Korea
Sarah Lehman, Temple University, USA
Kunal Lillaney, Amazon Web Services, USA
Enjie Liu, University of Bedfordshire, UK
Xiaodong Liu, Edinburgh Napier University, UK
Jay Lofstead, Sandia National Laboratories, USA
Hui Lu, Binghamton University (State University of New York), USA
Weibin Ma, University of Delaware, USA
Hosein Mohammadi Makrani, University of California, Davis, USA
Shaghayegh Mardani, University of California Los Angeles (UCLA), USA
Stefano Mariani, University of Modena and Reggio Emilia, Italy
Attila Csaba Marosi, Institute for Computer Science and Control - Hungarian Academy of Sciences, Hungary
Romolo Marotta, University of l'Aquila (UNIVAQ), Italy
Antonio Matencio Escolar, University West of Scotland, UK
Jean-Marc Menaud, IMT Atlantique, France
Philippe Merle, Inria, France
Nasro Min-Allah, Imam Abdulrahman Bin Faisal University (IAU), KSA
Preeti Mishra, Graphic Era Deemed to be University, Dehradun, India
Francesc D. Muñoz-Escoí, Universitat Politècnica de València, Spain
Ioannis Mytilinis, National Technical University of Athens, Greece
Hidemoto Nakada, National Institute of Advanced Industrial Science and Technology (AIST), Japan
Antonio Nehme, Birmingham City University, UK
Richard Neill, RN Technologies LLC, USA
Marco Netto, IBM Research, Brazil
Jens Nicolay, Vrije Universiteit Brussel, Belgium
Ridwan Rashid Noel, Texas Lutheran University, USA
Alexander Norta, Tallinn Technology University, Estonia
Aspen Olmsted, Simmons University, USA
Matthias Olzmann, noventum consulting GmbH - Münster, Germany
Brajendra Panda, University of Arkansas, USA
Lorena Parra, Universitat Politècnica de València, Spain
Arnab K. Paul, Oak Ridge National Laboratory, USA
Alessandro Pellegrini, National Research Council (CNR), Italy
Nancy Perrot, Orange Innovation, France
Tamas Pflanzner, University of Szeged, Hungary
Paulo Pires, Fluminense Federal University (UFF), Brazil
Agostino Poggi, Università degli Studi di Parma, Italy
Walter Priesnitz Filho, Federal University of Santa Maria, Rio Grande do Sul, Brazil
Abena Primo, Huston-Tillotson University, USA

Mohammed A Qadeer, Aligarh Muslim University, India
George Qiao, KLA, USA
Francesco Quaglia, University of Rome Tor Vergata, Italy
M. Mustafa Rafique, Rochester Institute of Technology, USA
Danda B. Rawat, Howard University, USA
Daniel A. Reed, University of Utah, USA
Christoph Reich, Hochschule Furtwangen University, Germany
Eduard Gibert Renart, Rutgers Unversity, USA
Ruben Ricart Sanchez, University West of Scotland, UK
Sashko Ristov, University of Innsbruck, Austria
Javier Rocher Morant, Universitat Politecnica de Valencia, Spain
Ivan Rodero, Rutgers University, USA
Takfarinas Saber, University College Dublin, Ireland
Rabia Saleem, University of Derby, UK
Hemanta Sapkota, University of Nevada - Reno, USA
Lutz Schubert, University of Ulm, Germany
Benjamin Schwaller, Sandia National Laboratories, USA
Savio Sciancalepore, TU Eindhoven, Netherlands
Wael Sellami, Higher Institute of Computer Sciences of Mahdia - ReDCAD laboratory, Tunisia
Jianchen Shan, Hofstra University, USA
Muhammad Abu Bakar Siddique, University of California, Riverside, USA
Altino Manuel Silva Sampaio, Escola Superior de Tecnologia e Gestão | Instituto Politécnico do Porto, Portugal
Alex Sim, Lawrence Berkeley National Laboratory, USA
Hui Song, SINTEF, Norway
Vasily Tarasov, IBM Research, USA
Zahir Tari, School of Computing Technologies | RMIT University, Australia
Bedir Tekinerdogan, Wageningen University, The Netherlands
Prashanth Thinakaran, Pennsylvania State University / Adobe Research, USA
Orazio Tomarchio, University of Catania, Italy
Reza Tourani, Saint Louis University, USA
Antonio Virdis, University of Pisa, Italy
Raul Valin Ferreiro, Fujitsu Laboratories of Europe, Spain
Massimo Villari, Università di Messina, Italy
Teng Wang, Oracle, USA
Hironori Washizaki, Waseda University, Japan
Mandy Weißbach, Martin Luther University of Halle-Wittenberg, Germany
Sebastian Werner, Information Systems Engineering (ISE) - TU Berlin, Germany
Magnus Westerlund, Arcada, Finland
Liuqing Yang, Columbia University in the City of New York, USA
Bo Yuan, University of Derby, UK
Christos Zaroliagis, CTI & University of Patras, Greece
Zhiming Zhao, University of Amsterdam, Netherlands
Jiang Zhou, Institute of Information Engineering - Chinese Academy of Sciences, China
Naweiluo Zhou, Höchstleistungsrechenzentrum Stuttgart (HLRS) - Universität Stuttgart, Germany
Hong Zhu, Oxford Brookes University, UK
Yue Zhu, IBM Research, USA

Jan Henrik Ziegeldorf, RWTH Aachen University, Germany
Wolf Zimmermann, Martin Luther University Halle-Wittenberg, Germany

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

An Automotive Penetration Testing Framework for IT-Security Education

Stefan Schönhärl, Philipp Fuxen, Julian Graf, Jonas Schmidt, Rudolf Hackenberg and Jürgen Mottok
Ostbayerische Technische Hochschule, Regensburg, Germany
Email:{stefan1.schoenhaerl, philipp.fuxen}@st.oth-regensburg.de
Email:{julian.graf, jonas.schmidt, rudolf.hackenberg, juergen.mottok}@oth-regensburg.de

***Abstract*—Automotive Original Equipment Manufacturer (OEM) and suppliers started shifting their focus towards the security of their connected electronic programmable products recently since cars used to be mainly mechanical products. However, this has changed due to the rising digitalization of vehicles. Security and functional safety have grown together and need to be addressed as a single issue, referred to as automotive security, in the following article. One way to accomplish security is automotive security education. The scientific contribution of this paper is to establish an Automotive Penetration Testing Education Platform (APTEP). It consists of three layers representing different attack points of a vehicle. The layers are the outer, inner, and core layers. Each of those contains multiple interfaces, such as Wireless Local Area Network (WLAN) or electric vehicle charging interfaces in the outer layer, message bus systems in the inner layer, and debug or diagnostic interfaces in the core layer. One implementation of APTEP is in a hardware case and as a virtual platform, referred to as the Automotive Network Security Case (ANSKo). The hardware case contains emulated control units and different communication protocols. The virtual platform uses Docker containers to provide a similar experience over the internet. Both offer two kinds of challenges. The first introduces users to a specific interface, while the second combines multiple interfaces, to a complex and realistic challenge. This concept is based on modern didactic theory, such as constructivism and problem-based learning. Computer Science students from the Ostbayerische Technische Hochschule (OTH) Regensburg experienced the challenges as part of a special topic course and provided positive feedback.**

Keywords*—*IT-Security; Education; Automotive; Penetration testing; Education framework.

I. Introduction

Automotive security is becoming increasingly important. While OEM have developed vehicles for a long time with safety as a central viewpoint, security only in recent years started becoming more than an afterthought. This can be explained by bringing to mind, that historically vehicles used to be mainly mechanical products. With the rising digitalization of vehicles, however, the circumstances have changed.

Recent security vulnerabilities based on web or cloud computing services, such as Log4j, can be seen as entry points into vehicles, which an attacker can use to cause significant harm to the vehicle or people. To combat this, the development and release of new standards are necessary. The International Organization for Standardization (ISO) 21434 standard and United Nations Economic Commission for Europe (UNECE) WP.29, show the importance of automotive security in recent years.

However, there are other ways in which automotive security can be improved. Jean-Claude Laprie defines means of attaining dependability and security in a computer system, one of these being fault prevention, which means to prevent the occurrence or introduction of faults [1]. This can be accomplished by educating current and future automotive software developers. Since vulnerabilities are often not caused by systemic issues, but rather programmers making mistakes, teaching them about common vulnerabilities and attack vectors, security can be improved. Former research shows furthermore that hands-on learning not only improves the learning experience of participants but also increases their knowledge lastingly. Therefore, a framework for IT-security education has been developed, which was derived from penetration tests on modern vehicles.

The ANSKo was developed as an implementation of this framework. It is a hardware case, in which communicating Electronic Control Unit (ECU)s are simulated, while their software contains deliberately placed vulnerabilities. In a first step, users are introduced to each vulnerability, before being tasked with exploiting them themselves.

This paper aims at establishing a realistic and effective learning platform for automotive security education. Therefore, the following research questions are answered:

- (RQ1) - What content is appropriate for an automotive penetration testing framework for IT-security education?
- (RQ2) - How could an automotive security education platform be implemented?

The structure of the paper starts with the related work in Section II. Section III introduces an architecture derived from modern vehicle technologies. Those technologies are then classified into layers and briefly explained in Section IV. The structure and used software of the ANSKo itself are presented in Section V. Section VI presents the learning concept and its roots in education theory. The paper ends with a conclusion in Section VII.

II. Related Work

Hack The Box (HTB) is a hands-on learning platform with several vulnerable virtual systems that can be attacked by the user. Thereby, a big focus of this platform is gamification.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

TABLE I
COMPARISON OF THE DIFFERENT APPROACHES

	HTB	HaHa SEP	RAMN	ANSKo
Virtual approach	YES	NO	NO	YES
Hardware approach	NO	YES	YES	YES
Automotive specific	NO	NO	YES	YES
Gamification	YES	NO	NO	YES
IT-Security	YES	YES	YES	YES

They do not offer automotive-specific systems and access to physical hardware is also not possible [2]. One approach that focuses on hardware-specific attacks is the Hardware Hacking Security Education Platform (HaHa SEP). It provides practical exploitation of a variety of hardware-based attacks on computer systems. The focus of HaHa SEP is on hardware security rather than automotive security. Students who are not present in the classroom can participate via an online course. A virtual version of the hardware cannot be used [3]. The Resistant Automotive Miniature Network (RAMN) includes automotive and hardware-related functions. The hardware is very abstract and is located on a credit card-sized Printed Circuit Board (PCB). It provides closed-loop simulation with the CARLA simulator but there is no way to use RAMN virtually. The focus of RAMN is to provide a testbed that can be used for education or research. However, it is not a pure education platform [4].

The fundamental and related work for the APTEP are real-world attack patterns. The technologies used for connected vehicles represent a particularly serious entry point into the vehicle, as no physical access is required. Once the attacker has gained access to the vehicle, he will attempt to penetrate further into the vehicle network until he reaches his goal. This can be done with a variety of goals in mind, such as stealing data, stealing the vehicle, or even taking control of the vehicle. The path along which the attacker moves is called the attack path. Such a path could be demonstrated, for example, in the paper ”Free-Fall: Hacking Tesla from wireless to Controller Area Network (CAN) Bus” by Keen Security Labs. The researchers succeeded in sending messages wirelessly to the vehicle’s CAN bus [5]. The same lab was also able to show further vulnerabilities, e.g., Bluetooth, Global System for Mobile Communications (GSM), and vehicle-specific services [6]. Valasek and Miller demonstrated the vulnerability of a vehicle’s infotainment system [7]. Using various attack paths, they managed to make significant changes to the vehicle.

Teaching at universities is often theory-based. As a result, many graduates may lack the practical experience to identify vulnerabilities. But it is precisely this experience that is of great importance in the professional field of software development, security testing, and engineering. The idea is to develop the competence level from a novice to an experts level, which can be guided by ”Security Tester” certified Tester Advanced Level Syllabus. The described APTEP presents an ecosystem to establish such learning arrangements in which constructivism-based learning will happen [8][9].

III. ARCHITECTURE

The attacks from the previous section show, that attacks follow a similar pattern. There is an entry point through which the attacker gains access to the vehicle. He then tries to move through the vehicle network by exploiting further vulnerabilities. He does this until he reaches his target. To represent this procedure in the architecture of ANSKo, it was divided into different layers.

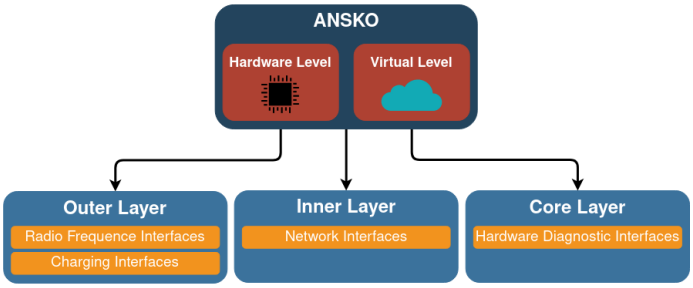


Fig. 1. ANSKO Architecture

As shown in Figure 1, the following three layers were chosen: Outer layer, inner layer, and core layer. They delimit the respective contained interfaces from each other.

A. Outer Layer

The automotive industry is currently focusing heavily on topics, such as automated driving functions, Vehicle-to-Everything (V2X) networking, and Zero-Emission Vehicles (ZEV). In these areas, new trend technologies can lead to valuable new creations. But unfortunately, this development also favors the emergence of new and more critical points of attack. For this reason, the outer layer was included in the APTEP as part of the architecture. It contains all the functionalities that enable the vehicle to communicate with its environment. This includes the two V2X technologies Cellular-V2X and WLAN-V2X as well as other communication protocols, such as Bluetooth and GSM. In addition to the communication protocols, there are also interfaces, such as various charging interfaces, sensors, and much more.

The outer layer represents an important component because many interfaces contained in it represent a popular entry point for attacks. This is the case because the technologies used there are usually an option to potentially gain access to the vehicle without having physical access to the vehicle. Even if the sole exploitation of a vulnerability within the outer layer does not always lead to direct damage in practice, further attack paths can be found over it. In most cases, several vulnerabilities in different areas of the vehicle system are combined to create a critical damage scenario from the threat. Therefore, vehicle developers need to be particularly well trained in this area.

B. Inner Layer

The inner layer of the APTEP represents the communication between individual components. While modern vehicles

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

implement different forms of communication, bus systems like Controller Area Network (CAN), Local Interconnect Network (LIN), and FlexRay used to be predominant. Since modern vehicle functions connected to the Outer Layer, like image processing for rearview cameras or emergency braking assistants [10], require data rates not achievable by the previously mentioned bus systems, new communication systems, like Ethernet, have been implemented in vehicles.

Depending on the scope, the mentioned bus systems are still in use because of their low cost and real-time capabilities. From those communication technologies, different network topologies can be assembled. Individual subsystems connecting smaller components, e.g., ECUs, are themselves connected through a so-called backbone. Gateways are implemented to connect the subsystems with the backbone securely.

After gaining access to a vehicle through other means, the inner layer represents an important target for attackers since it can be used to manipulate and control other connected components. While the target components can be part of the same subsystem, it is also possible, that it is part of a different subsystem, forcing the attacker to communicate over the backbone and the connected gateways. The inner layer thus represents the interface between the outer - and core layer.

C. Core Layer

Manipulating the ECUs of a vehicle themselves results in the greatest potential damage and therefore represents the best target for a hacker. In the APTEP, this is represented as the core layer.

Vehicles utilize ECUs in different ways, e.g., as a Body Control Module, Climate Control Module, Engine Control Module, Infotainment Control Unit, Telematic Control Unit. In addition, electric vehicles include further ECUs for special tasks, such as charging.

If attacks on an ECU are possible, its function can be manipulated directly. Debugging and diagnostic interfaces, like Joint Test Action Group (JTAG) or UDS (Unified Diagnostic Services), are especially crucial targets since they provide functions for modifying data in memory and reprogramming of ECU firmware.

The impact of arbitrary code execution on an ECU is dependent on that ECUs function. While taking over, e.g., a car's infotainment ECU should only have a minor impact on passengers' safety, it can be used to attack further connected devices, via inner layer, from an authenticated source. The goal of such attack chains is to access ECUs where safety-critical damage can be caused. Especially internal ECUs interacting with the engine can cause severe damage, like shutting off the engine or causing the vehicle to accelerate involuntarily.

IV. Interfaces

This section describes some chosen interfaces of the previously presented layers. The selection was made from the following three categories: "Radio Frequency and Charging Interfaces", "Network Interfaces" and "Hardware Diagnostic Interfaces".

Implemented in the ANSKo is one interface from each architecture layer - NFC from the outer layer (Section IV-A1c, CAN from the inner layer (Section IV-B1), and UDS from the core layer (Section IV-C2). This facilitates the cross-domain challenges described in Section VI.

A. Radio Frequency and Charging Interfaces

The outer layer contains the interfaces of the category "radio frequency and charging interfaces". They all have in common that they enable the vehicle to communicate with its environment. Furthermore, the included interfaces can be divided into the following classes: short-range communication, long-range communication, and charging interfaces.

1) Short-range Communication:

a) Bluetooth: Bluetooth is a radio standard that was developed to transmit data over short distance wireless. In the vehicle, the radio standard is used primarily in the multimedia area. A well-known application would be, for example, the connection of the smartphone to play music on the vehicle's internal music system.

b) RFID: Radio frequency identification (RFID) enables the communication between an unpowered tag and a powered reader. A powered tag makes it possible to increase the readout distance. RFID is used, for example, in-vehicle keys to enable keyless access.

c) NFC: Near field communication (NFC) is an international transmission standard based on RFID. The card emulation mode is different from RFID. It enables the reader to also function as a tag. In peer-to-peer mode, data transfer between two NFC devices is also possible. In vehicles, NFC is used in digital key solutions.

d) WLAN-V2X: The WLAN-V2X technology is based on the classic WLAN 802.11 standard, which is to be used in short-range communication for V2X applications. However, almost all car manufacturers tend to focus on Cellular-V2X because long-range communication is also possible in addition to short-range communication.

2) Long-range Communication:

a) GNSS: The Global Navigation Satellite System (GNSS) comprises various satellite navigation systems, such as the Global Positioning System (GPS), Galileo, or Beidou. Their satellites communicate an exact position and time using radio codes. In vehicles, GNSS is mainly used in onboard navigation systems. Furthermore, it is increasingly used to manage country-specific services.

b) Cellular-V2X: Cellular-V2X forms the communication basis for V2X applications. It uses the cellular network for this purpose. In contrast to WLAN-V2X, it enables both V2V and vehicle-to-network (V2N) communication.

3) Charging Interfaces: To enable charging or communication between an electric vehicle and a charging station, a charging interface is required. Due to the high diversity in this area, there is not just one standard.

a) CHAdeMO: The CHAdeMO charging interface was developed in Japan where it is also used. The charging process can be carried out with direct current (DC) charging. Mainly

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

Japanese OEMs install this charging standard in their vehicles. Some other manufacturers offer retrofit solutions or adapters.

b) Tesla: Tesla predominantly uses their own charging interface, which allows both alternating current (AC) and DC charging. However, due to the 2014/94 EU standard, Tesla is switching to the Combined Charging System (CCS) Type-2 connector face in Europe.

c) CCS: The official European charging interfaces CSS Type-1 and CSS Type-2 are based on the AC Type-1 and Type-2 connectors. The further development enables a high DC charging capacity in addition to the AC charging.

B. Network Interfaces

Network interfaces describe the technologies used to communicate between components, like ECUs or sensors. It represents the inner layer.

1) CAN: CAN is a low-cost bus system, that was developed in 1983 by Bosch. Today it is one of the most used bus systems in cars since it allows acceptable data rates of up to 1Mbit/s while still providing real-time capabilities because of its message prioritization. Its design as a two-wire system also makes it resistant to electromagnetic interference.

Traditionally in a vehicle CAN is often used as the backbone, providing a connection between the different subsystems. It is also used in different subsystems itself, like engine control and transmission electronics.

2) LIN: The LIN protocol was developed as a cost-effective alternative to the CAN bus. It is composed of multiple slave nodes, which are controlled by one master node, which results in a data rate of up to 20Kbit/s.

The comparatively low data rate and little fault resistance result that LIN being mainly used in non-critical systems, like power seat adjustment, windshield wipers, and mirror adjustment.

3) MOST: The Media Oriented System Transport (MOST) bus provides high data rates of 25, 50, or 150 Mbit/s depending on the used standard. It was developed specifically for use in vehicles and is typically implemented as a ring.

As the name suggests the field of application for the MOST bus is not in safety-critical systems. but in multimedia systems of a vehicle. Since transmission of uncompressed audio and video data requires high data rates, MOST are suited best for those tasks.

4) FlexRay: FlexRay offers data transmission over two channels with 10Mbit/s each. They can be used independently or by transmitting redundant data for fault tolerance. Furthermore, FlexRay implements real-time capabilities for safety-critical systems.

FlexRay was developed with future X-by-Wire (steer, brake, et al.) technologies in mind [11]. Even though FlexRay and CAN share large parts of their requirements, FlexRay improves upon many aspects, leading to it being used as a backbone, in powertrain and chassis ECUs and other safety-critical subsystems.

5) Ethernet: Automotive Ethernet provides a cost-effective transmission protocol with high data rates of 1Gbit/s. While the underlying Ethernet protocol is not fit to be used in systems with electromagnetic interference and also offers no real-time capabilities, this can be remedied by using the BroadR-Reach and Audio-Video-Bridging (AVB) standards respectively.

Due to the constant increase in required data rates in new technologies, such as image processing, Ethernet was adapted for its use in vehicles. Because of its widespread use even outside of vehicles, it offers many different protocols, which are constantly being improved.

C. Hardware-Diagnostic Interfaces

The hardware-diagnostic interfaces are classified in the core layer. They describe technologies, that allow interaction between a person, such as a programmer, and an ECU to allow, e.g., reprogramming of the software.

1) Debug: Debug interfaces are used in embedded development to allow debugging, reprogramming, and reading out error memory of the circuit boards. Vehicles implement various debug interfaces, depending on their integrated circuit boards. The most common interfaces include Joint Test Action Group (JTAG), Serial Wire Debug (SWD), Universal Asynchronous Receiver Transmitter (UART), and Universal Serial Bus (USB).

Interacting with the debug interfaces requires special equipment, like adapters.

2) UDS: Modern vehicles implement a diagnostic port as well to allow independent car dealerships and workshops functionalities similar to the debug interfaces while not being unique to one particular OEM. It uses the communication protocol Unified Diagnostic Services (UDS), defined in the ISO 14229 standard.

UDS utilizes CAN as the underlying protocol to transmit messages. To prevent unauthorized access to the diagnostic port, UDS provides different tools, like ”Diagnostic Session Control” which defines different sessions, such as default, diagnostic, or programming. OEMs can choose which service is available in each session. Security-critical services can also be further guarded by using the ”Security Access” which protects the respective service through a key seed algorithm.

3) Side Channels: The final interface in the core layer are side channels. A computing unit emits certain side-channel data while performing operations, such as the consumed energy while encrypting data. They allow attackers to gain information about secret parts of the computer system like the used keys for cryptographic operations. Side-channel data can therefore be used to attack otherwise secure computer systems. Possible different side channels include time, power, fields, and temperature.

V. Structure

The presented APTEP is implemented in the ANSKo, which consists of a hardware and a virtual level. Their required components and used software are described in the following.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

A. Hardware-Level

The goal of the ANSKo is to provide a low-cost learning environment for automotive security. A picture of the hardware

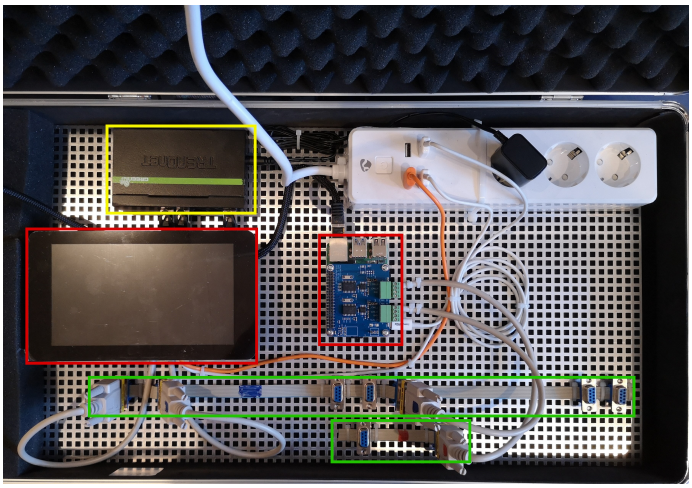


Fig. 2. ANSKo Hardware

contents can be seen in Figure 2. The currently included components are marked by colors. It is intended to further extend the platform by the listed interfaces in Section IV.

- **Yellow - Ethernet Switch:** The Ethernet switch connects to both Raspberry Pis and allows additional connection to the user.
- **Red - Display and Raspberry Pis:** The main components of the case are two Raspberry Pis, which simulate ECUs in a vehicle. They possess a PiCAN Duo board allowing two independent CAN connections. One of the Raspberry Pis possesses a display, simulating a dashboard with a speedometer and other vehicle-specific values.
- **Green - CAN Bus:** The CAN Bus is the main communication channel in the current structure. Connected devices can be disconnected by removing the respective cables.

One implemented challenge in the ANSko is a Man-in-the-Middle attack. The goal is to lower the displayed mileage of the car to increase its value. A user working with the ANSKo needs to read the messages being sent between the simulated ECUs. They can interact with the CAN Bus by connecting to the CAN Bus via USB cable and an included Embedded60 microcontroller.

The operating system running on the Raspberry Pis was built by using pi-gen It allows generating and configuring a Raspberry Pi OS image. By using the automation software Ansible, challenges can be installed on all cases simultaneously. Challenges are started as a systemd service after copying the required files to the cases.

B. Virtual-Level

During the Covid-19 pandemic holding education courses hands-on was not possible. To still provide the advantages of the ANSKo during lockdowns, an online platform with identical challenges has been realized.

The virtual challenges are accessible through a website, which allows the authentication of users. A user can start a challenge, which creates a Docker container. This ensures an independent environment for users while also protecting the host system. Users can receive the necessary CAN messages by using the socketcand package, providing access to CAN interfaces via Transmission Control Protocol/Internet Protocol (TCP/IP).

The unique docker containers for each user allow them to stop and start working on the challenge at any time but limits the maximum amount of users attempting the challenges concurrently. Validation of a correct solution also does not have to be carried out manually by sending a unique string of characters on the CAN bus which can be compared to the back end by the user.

VI. Learning Concept

ANSKo's concept of learning is based on the theory of constructivism. It allows learners to achieve the higher-order learning goals of Bloom's Taxonomy. They are more capable of analyzing facts and problems, synthesizing known information, and evaluating their findings [12].

Learning concepts are used to encourage learners to actively think rather than passively absorb knowledge, e.g., Problem-Based Learning (PBL). ANSKo consists of several real-world problems, so-called challenges. Support for problem-solving uses the scaffolding approach, i.e., learners initially receive theoretical knowledge, optimize their learning progress in groups, and solve the problem independently [12].

The challenges can be divided into two categories: "Domain-specific challenges" and "Cross-domain challenges". The two types each pursue different learning objectives.

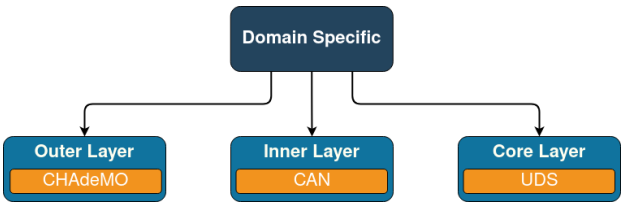


Fig. 3. Domain-specific Challenge

As shown in Figure 3, "Domain-specific challenges" are about learning the functionalities and vulnerabilities of a single interface within a domain. A challenge is considered complete when the learner has found and exploited the vulnerability.

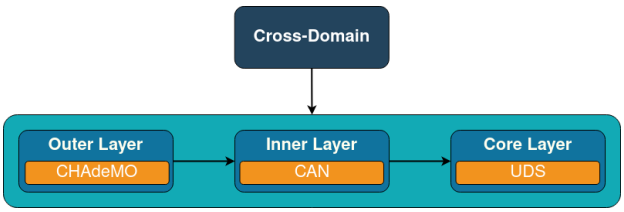


Fig. 4. Cross-domain Challenge

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

Cross-domain challenges aim to teach the learner how to find and exploit attack paths. Figure 4 shows an example of a cross-domain challenge. Here, interfaces from the different layers are combined. The difficulty level of these challenges is higher and therefore the respective domain-specific challenges for the required interfaces have to be solved first.

Computer science students from the OTH Regensburg were able to work with the ANSKo as part of a special topic course for the 6th & 7th semesters. The course evaluation, which was answered by the students, showed the benefit of the learning platform. They reported a positive experience when working with the ANSKo, e.g., when asked about understanding the importance of automotive security or their learning progress. The selected challenges were quoted as adequately difficult to be solved using the underlying learning concept.

VII. Conclusion

The presented vulnerabilities at the beginning of this paper and the listing of strengths and weaknesses of existing learning platforms justify the need for an automotive-specific IT security learning platform. For this reason, an APTEP was developed on which participants can learn about vulnerabilities in practice.

To realize this, an architecture for the APTEP was chosen that maps the described attacks. The architecture consists of three layers - outer layer, inner layer, and core layer. Each of them contains different interfaces, such as the Radio Frequency interface as well as the Charging interface in the outer layer, Network interfaces in the inner layer, and Hardware-Diagnostic interfaces in the core layer.

The APTEP is implemented on the Hardware level to provide a realistic learning environment, but also offers a virtual level, which allows users to work with the platform remotely since the Covid-19 pandemic prevented hands-on work.

To keep the challenges as realistic as possible while providing learners with an appropriate level of complexity, the tasks were divided into two categories. There are ”Domain-specific challenges,” which deal with only one interface per challenge. A ”Cross-domain challenge” cannot be solved until the associated ”Domain-specific challenges” have been solved for each included interface. The ”Cross-domain challenges” combine different interfaces and teach learners to find and exploit attack paths.

Future work includes the implementation of electric vehicle-specific challenges, e.g., charging interfaces. Side-channel attack challenges will be included as well.

To support the individual learning progress eye tracking will be included and analyzed. The learner's cognitive load will be determined by AI-based classification results. Finally, this will improve individual learning success.

References

[1] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” 2004.

[2] H. the Box, *Hack the box*. [Online]. Available: https://www.hackthebox.com/ (retrieved: 02/2022).

[3] S. Yang, S. D. Paul, and S. Bhunia, “Hands-on learning of hardware and systems security.,” *Advances in Engineering Education*, vol. 9, no. 2, n2, 2021. [Online]. Available: https://files.eric.ed.gov/fulltext/EJ1309224.pdf (retrieved: 02/2022).

[4] C. Gay, T. Toyama, and H. Oguma, “Resistant automotive miniature network,” [Online]. Available: https://fahrplan.events.ccc.de/rc3/2020/Fahrplan/system/event_attachments/attachments/000/004/219/original/RAMN.pdf (retrieved: 02/2022).

[5] S. Nie, L. Liu, and Y. Du, “Free-fall: Hacking tesla from wireless to can bus,” *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017. [Online]. Available: https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf (retrieved: 02/2022).

[6] Z. Cai, A. Wang, W. Zhang, M. Gruffke, and H. Schweppe, “0-days & mitigations: Roadways to exploit and secure connected bmw cars,” *Black Hat USA*, vol. 2019, p. 39, 2019. [Online]. Available: https://i.blackhat.com/USA-19/Thursday/us-19-Cai-0-Days-And-Mitigations-Roadways-To-Exploit-And-Secure-Connected-BMW-Cars-wp.pdf (retrieved: 02/2022).

[7] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, no. S 91, 2015.

[8] F. Simon, J. Grossmann, C. A. Graf, J. Mottok, and M. A. Schneider, *Basiswissen Sicherheitstests: Aus- und Weiterbildung zum ISTQB® Advanced Level Specialist – Certified Security Tester*. dpunkt.verlag, 2019.

[9] International Software Testing Qualifications Board, *Certified tester advanced level syllabus security tester, international software testing qualifications board*, 2016. [Online]. Available: https://www.german-testing-board.info/wp-content/uploads/2020/12/ISTQB-CTAL-SEC_Syllabus_V2016_EN.pdf (retrieved: 02/2022).

[10] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, “Automotive ethernet: In-vehicle networking and smart mobility,” in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pp. 1735–1739. DOI: 10.7873/DATE.2013.349.

[11] W. Zimmermann and R. Schmidgall, *Busssysteme in der Fahrzeugtechnik [Bus systems in automotive engineering]*, ger. Springer Vieweg, 2014, p. 96.

[12] G. Macke, U. Hanke, W. Raether, and P. Viehmann-Schweizer, *Kompetenzorientierte Hochschuldidaktik [Competence-oriented university didactics]*, ger, 3rd ed. Beltz Verlagsgruppe, 2016, ISBN: 9783407294852. [Online]. Available: https://content-select.com/de/portal/media/view/56cc0a3a-741c-4bd7-8eab-5eeeb0dd2d03 (retrieved: 03/2022).

Design and Implementation of an Intelligent and Model-based Intrusion Detection System for IoT Networks

Peter Vogl, Sergei Weber, Julian Graf, Katrin Neubauer, Rudolf Hackenberg
Ostbayerische Technische Hochschule, Regensburg, Germany
Dept. Computer Science and Mathematics
email:{peter.vogl, sergei.weber, julian.graf, katrin1.neubauer, rudolf.hackenberg}@oth-regensburg.de

***Abstract*—The ongoing digitization and digitalization entails the increasing risk of privacy breaches through cyber attacks. Internet of Things (IoT) environments often contain devices monitoring sensitive data such as vital signs, movement or surveillance data. Unfortunately, many of these devices provide limited security features. The purpose of this paper is to investigate how artificial intelligence and static analysis can be implemented in practice-oriented intelligent Intrusion Detection Systems to monitor IoT networks. In addition, the question of how static and dynamic methods can be developed and combined to improve network attack detection is discussed. The implementation concept is based on a layer-based architecture with a modular deployment of classical security analysis and modern artificial intelligent methods. To extract important features from the IoT network data a time-based approach has been developed. Combined with network metadata these features enhance the performance of the artificial intelligence driven anomaly detection and attack classification. The paper demonstrates that artificial intelligence and static analysis methods can be combined in an intelligent Intrusion Detection System to improve the security of IoT environments.**

***Keywords*—Intrusion Detection; Artificial Intelligence; Machine Learning; Network Security; Internet of Things.**

I. Introduction

Demographic change is a particular challenge worldwide. One consequence of demographic change is an aging population. However, because of the now higher life expectancy, the risk of illness for each older person is also increasing [1]. For this reason, measures must be taken to enable the aging population to live more safely.

To imporve safetiness Ambient Assisted Living (AAL) is used. AAL refers to all concepts, products and services that have the goal of increasing the quality of life, especially in old age, through new technologies in everyday life [2]. The intelligent Intrusion Detection System (iIDS) described in this paper is part of a publicly funded research project Secure Gateway Service for Ambient Assisted Living (SEGAL). Within SEGAL, a lot of sensitive information such as heart rates, blood sugar or blood pressure are measured. These are needed so that people in need of care can live in their familiar environment for as long as possible. This is made possible by developing an AAL service for SEGAL. Within the AAL service, the recorded data from Internet of Things (IoT) devices are sent via the smart meter gateway to the AAL data management of the responsible control center from the AAL-Hub. The smart meter gateway is a secure communication channel, as a certificated communication path is used for the transmission of the recorded data [3]. However, the exchange of data between the IoT devices and the AAL hub is not necessarily to be considered secure and can be seen as a target for attacks. Therefore, there is the need to secure the communication between the IoT devices and the back end system, so that there is no theft and manipulation of the transmitted data. In this case, the iIDS is used to protect the sensitive recorded data, as it is intended to detect possible attacks. The individual layers of the iIDS are designed to be easily integrated into cloud structures. This allows cloud to take advantage of flexibility and efficiency to monitor network security optimally. Therefore, security services can be scaled depending on the circumstances [4]. In addition, the cloud offers the possibility to improve new innovative Artificial Intelligence (AI) security analytics and adapt them to the supervision of different networks.

This paper is a continuation of [5], in which the architecture of the iIDS has already been presented in detail. The implementation of the intelligent and model-based iIDS, including the explanation of attack detection methods is shown in this paper.

The structure of this paper is organized as follows: Section II describes the related work. Section III presents the architecture of the iIDS. In Section IV the rule-based modules of iIDS are described in detail. Section V deals with the explorative data analysis, while Section VI describes data preprocessing which is required for the AI modules. The used AI based modules of the iIDS are shown in Section VII, followed by a conclusion and an outlook on future work VIII.

II. Related Work

In recent years, AI methods have been increasingly used in many different sectors including the healthcare sector. The increasing number of IoT networks needs to be detected reliably and conscientiously from cyber attacks. Different approaches are used for the respective iIDS. Vinayakamur et. al [6] are using self-taught learning as a deep learning approach. Two steps are required to detect attacks. To begin with, the feature representation is learned from a large collection of

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

unlabelled data. Subsequently, this learned representation is applied to labelled data and thus used for the detection of attacks. Summerville et. al [7] use anomaly detection as their main detection method. The basis is a deep packet analysis approach, which uses a bit-pattern technique. The network payload is described as a sequence of bytes, also called bit-pattern. Feature extraction is done as an overlapping tuple of bytes, also known as n-grams. McDermott et. al [8], on the other hand, use a machine learning approach to detect botnets in IoT networks. They developed a model based on deep bidirectional Long Short Term Memory using a Recurrent Neural Network. Burn et. al [9] are using a deep learning approach for detecting attacks, in which they use a dense random neural network.

The approach for the iIDS differs in some aspects. On the one hand, we use common network analysing methods further described as static methods and on the other hand we use state of the art AI aproaches to detect anomalies and classify attacks. The previous mentioned approaches can detect anomalies, but none of them can classify attacks. It is also our goal to achieve a zero false positive rate by using AI algorithms and the static based models. As mentioned in our previous paper, we still do not know of a familiar combination that uses the same AI algorithms combined with the static based models.

Therefore two major research questions are to be answered in this paper:

- **RQ 1:** Can artificial intelligence and static analysis be sustainable implemented in practice-oriented intelligent Intrusion Detection System?
- **RQ 2:** How can static and dynamic methods be developed and combined to improve network attack detection?

The goal of this paper is to answer the identified research questions by presenting procedures and techniques for achieving advanced network monitoring.

III. Architecture

The architecture of the iIDS consists of 5 layers, with a Data Collection Layer (DCL) as it's foundation and a Reaction Layer as top layer. The organization of the individual layers and their connections are shown in detail in Figure 1.

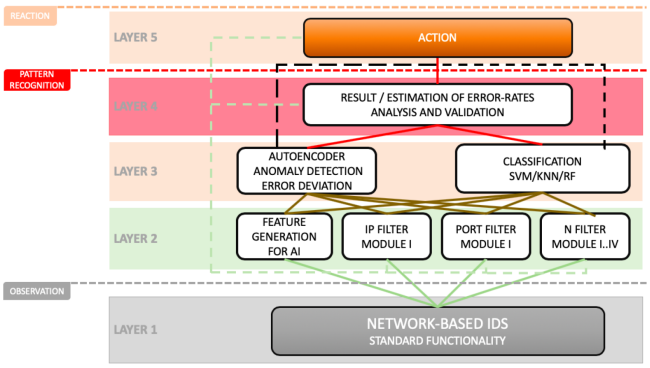


Figure 1. Architecture of the Intelligent Intrusion Detection System [5]

The DCL implements the capturing and conversion mechanism to monitor network traffic and to extract the required transmission information. All data is also stored in a MySQL database for later usage. On top of the DCL, several rule-based modules are implemented to analyse and filter probably malicious traffic with static network observation methods. Also, the data preparation for the upcoming machine learning-based modules is part of this section. The third layer locates the different AI modules used to detect intrusions and to classify the type of attack. For example, a neural network module for anomaly detection is realized at this point of the architecture. A deeper insight into these methods will be given in Section VII. All our modules, rule-based and AI-based, are designed to return an assessment over their predicted outcome. In the penultimate layer, all the return values are evaluated and the probability of an intrusion will be calculated. Based on this calculation and through additional information for example, from the classifier in the third layer the last layer can deploy dedicated security actions to prevent or limit damage to the system. Possible countermeasures could be notifications to an administrator, the shutdown of a connected device, or the interruption of the Internet connection as a final action.

To get a light weighted and expandable system, all major components, like the iIDS itself, the AI-based modules, or the MySQL database, are deployed in their own Docker containers and can be managed independently.

IV. Rule-based Modules

As mentioned in Section III, the rule-based modules are part of the second layer in the presented architecture. They act as a first security barrier and are capable to give roughly feedback about security issues based on the port and address information of Layer 2 and 3 of the ISO/OSI network model.

A. Analysing Port Information

Two different modules are implemented to analyse the network's port information. The first one allows monitoring the individual port usage. With an analysis of the network packages, the commonly used ports of the network participants can be discovered, which enables the ability to whitelist these ports. In reverse, packages which do not have at least a whitelisted source or destination port number will be treated as a possible malicious package and an intrusion assessment value for the subsequent evaluation will be addressed to the next layer. The second module is designed to discover port scan attacks. The purpose of a port scan is to evaluate the open ports of a target system which can be used to set up a connection. Despite a port scan is not an illegal action, at least in Germany, it is often used to get information about a target for later attacks. Because of this common intention and their easy execution with open-source software like Nmap [10], port scans will be treated as a threat indicator for the iIDS.

B. Analysing Address Information

Part of the captured data from the Data Link Layer and the Network Layer is the address information. Based on the unique

MAC-Address and the allocation of a static IP the trusted network members can be verified. To further enhance security also the state of the dynamic host configuration protocol is analysed for violations of thresholds, such as IP range limits. The obtained information is also used to support the AI-based modules and provides important indicators for the reaction layer to defend against attacks.

V. Explorative Data Analysis

Explorative Data Analysis (EDA) provides a statistic insight into a given data set, enables the recognition and visualisation of dependencies and anomalies, and forms the basis for further feature extractions [11].

A. Data Insights

The used data set for training and testing the AI-based modules is based on a laboratory replica of a smart home (SHLab) that delivers network data from common IoT devices. Table I shows the scope of the used data set based on different labels. Two-thirds of the data are packages from normal daily data traffic, one-third are attack packages. Most of the malicious data are Distributed Denial of Service (DDoS) attacks, divided into SYN-, PSH-ACK-, FIN-, ICMP- or UDP-floods, but also Wi-Fi-Deauthentication attacks are included.

TABLE I
Composition of the Used Data Set

Intrusion Class	Packages
Normal Data	**908355**
Wi-Fi-Deauthentication Attack	**32049**
DDoS Attacks	**468769**
— SYN-Flood	147849
— FIN-Flood	27408
— PSH-ACK-Flood	20971
— ICMP-Flood	185058
— UDP-Flood	87483
Combined Dataset	**1409173**

The focus of the iIDS is on the metadata analysis of the header information. The payload information is also only captured as metadata because it is often transmitted in encrypted form. Overall, 52 different data features are captured from the OSI layers 2, 3 and 4. This includes the address and port information mentioned in IV and furthermore data from the Transport Layer, for example, the TCP flags or checksums.

A correlation analysis allows a better insight into the correlations between important features. How well the features fit together is indicated by the correlation coefficient. The coefficient scales from – 1 to 1, whereby a value of 0 indicates no correlation between two features and -1 and 1 both indicate a strong linear correlation. Figure 2 shows a correlation matrix of the most important metadata.

One of these important features is the packet length. A comparison of normal data and attack data showed that attack packages have a significant lower packet length. Furthermore, the evaluation revealed that the attacks don't change the packet length over the attack time span. Another feature is the data offset of TCP packages, which is an indicator for the header

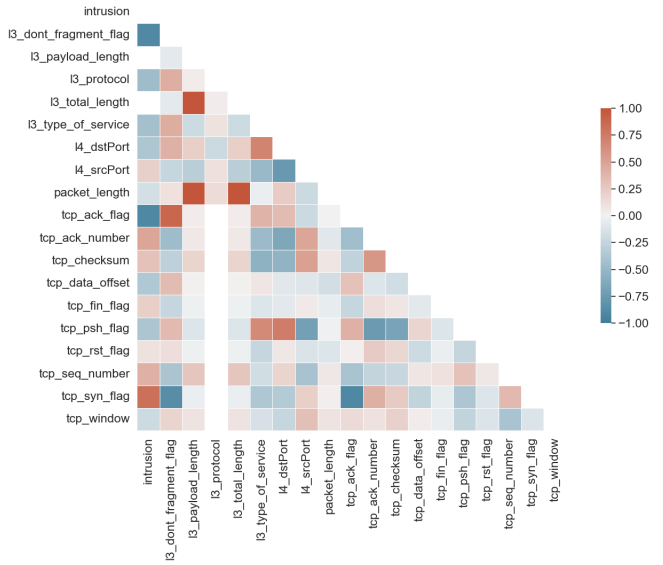


Figure 2. Feature Correlations

size because it contains the position of the payload in a packet. In addition to a shorter packet length, a more detailed insight showed that attack packages also have a shorter header length, which leads to a smaller data offset. DDoS attacks aim to flood their target with a large number of packages. To achieve this, it is useful to have the bare minimum of packet size. This contains a small payload size and the least amount of header options and as a result, a small data offset. These and several additional network characteristics, such as port, flag, protocol information, are examined in order to be able to derive sustainable input for the iIDS modules.

B. Feature Extratction

For the extraction of new features from the data set two different concepts were developed. Both approaches derive additional information from the temporal context of the network packets. However, a distinction is made here as to when or to what extent network packets are measured at the node. In both processing methods, the incoming network packets are aggregated into small blocks of different characteristics further called as block-based approach. Hereby the data is ether combined to a specific number of packets measured on the order of arrival on the node or measured on passing the node in specific time windows.

Quantity-Blocks: Combines the data captured by the observation level of the iIDS to equal sized blocks calculated on a specific count value.

Time-Window-Blocks: Combines the data captured by the observation level of the iIDS to equal sized blocks based on fixed time frames.

However, the best results have been achieved by combining both concepts together. To detected network attacks the incoming packets of each local network member is separated by destination IP or MAC addresses. These packets are then processed by both methods and combined to quantity- and

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

time-based blocks. This allows a device-specific analysis of the network traffic (Figure 3). Through this combined concept

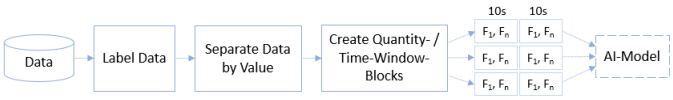


Figure 3. Feature Generation Process

time-based correlation can be used for each device to extract additional features to enhance AI-based attack pattern recognition. This modern, unconventional approach makes it possible to find new classification patterns and integrate them into the analyzes of the iIDS.

VI. Data Preprocessing

Preprocessing the data is also performed on the second layer alonge the rule-based modules. Data preprocessing consists of cleaning, labeling, encoding, normalization and standardisation of the caputred data.

A. Encoding

The encoding of the captured package data is an important step for later usage. Some network information like the MAC or IP addresses but also the different protocol types like TCP, UDP or ICMP are stored in a MySQL VARCHAR data format. Another already mentioned reason for encoding parts of the data is to make it accessible for the AI modules. The majority of ML models require specific data types.

There were two options to cope with that problem. The first one was to exclude this data from later usage. This is not a suitable option because of the importance of the information for the classification. To make the information usable for the AI-based modules a label encoding function is used. Label encoding replaces the distinct categories, i.e. the unique MAC addresses, with a numeric value. Through this, the specific value gets lost, but the overall correlation is still intact, which is important for the ML usage.

B. Cleaning

Missing data and NaN values are an additional problem for most ML models. Due to the huge variety of protocols used in network traffic the entries in the data set often contains empty fields. In example, an IPv4 package has no specific IPv6 information and vice versa. Features with more than 20% missing data entries are removed from the data set, because no statistical interpolation parameters can be calculated from the limited data stock, which can fill the missing gaps without errors. This doesn't apply for all network values like e.g. TCP flags. Therefore and for the other features we use different interpolation methods based on the specification of the feature to deal with missing data. This includes the use of mean and median interpolation for the empty data fields.

C. Normalization and Standardisation

Feature rescaling is an often-done step in the data preprocessing phase of most AI-based models. It is not an essential requirement and not all algorithms benefit in the same way from this process. However, it can lead to better learning performance because most AI-based models perform poorly if the input features have significant different scales. Features with a bigger scale have more weight in each iteration and subsequently dominate the training process. Through rescaling, this disparity in weighting between the features should be minimized.

There are two major ways to perform feature rescaling: normalization and standardisation. The normalization, also often called min-max-scaling, converts the original range of individual features to a general scale for all features. A common interval for this scale is [0, 1] [12, p. 44]. Figure 4 shows on the left side the original range of the port numbers with a scale of 0 to 65535. On the right site the port numbers have been normalized with a Min-Max-Scaler.

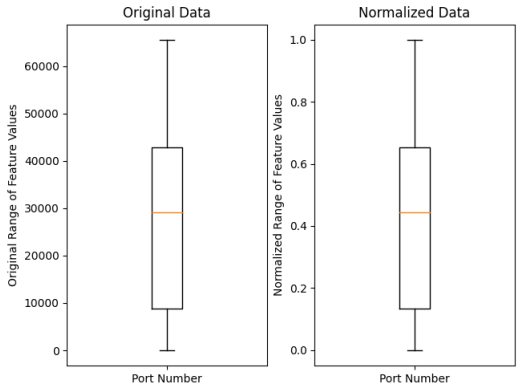


Figure 4. Comparison of Original and Normalized Data

The standardisation shapes the feature values in the proportion of a normal distribution. The mean value of the normal distribution is calculated over the elements of a feature. Unlike normalization, the interval limits are not given values. However, the standard deviation from the mean value is used to set the scale for the feature rescaling. Standardisation is often used for data with a natural standard distribution [12, p. 44].

D. Snorkel

Snorkel is a python package for labelling AI training data based on the work of Alex Ratner [13]. The proposed solution is, to enable the developer to implement labelling functions, which programmatically imply rules to label the data.

The implementation process starts by aggregating the data over 10 second time intervals. As already mentioned in Section V-B this is necessary to improve the performance of the AI-based modules and Snorkel is able to benefit from this procedure too. The aggregated data is used to generate specific indices for each intrusion class. Most flooding attacks don't change their parameters during an attack, therefore the

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

assumption is that the most common parameter subsets belong to flood packages. For a TCP flood, this leads to an index with the destination port and the packet length as parameters. This approach is also flexible enough to handle continuous new data from the SHLab without the need for changes. Trained on the data set mentioned in Section V-A Snorkel is able to classify all aggregated entries with an accuracy of 90-95%.

The difficulty is to find a classification model with two important properties. The first requirement is a good training result with the aggregated and labelled data delivered from Snorkel. The second requirement is a high accuracy on normal network data delivered from the SHLab. To test these requirements different classification models are used, like a Nearest Neighbour Model, a Support Vector Machine, a Logistic Regression Model, a Decision Tree and a Random Forest. All models accomplish the first requirement with an average accuracy of 90%. The second requirement is more difficult for some models. The Nearest Neighbour Model and the Support Vector Machine achieve 10% accuracy, followed by the Logistic Regression Model (75%) and the Decision Tree (85%) with noticeably better results. The Random Forest fulfilled both requirements with 90% accuracy.

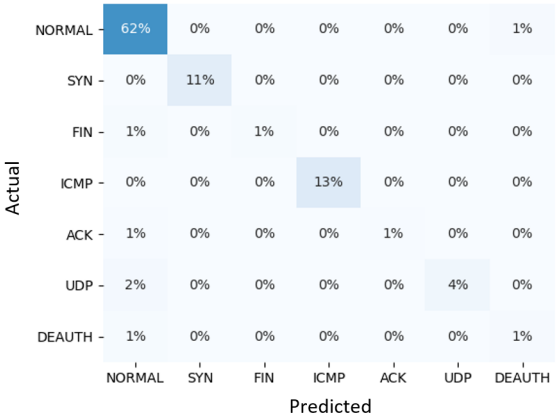


Figure 5. Confusion Matrix of the Random Forest Model

Figure 5 shows the detailed result of the Random Forest in form of a confusion matrix with the performed attacks and the distribution of the actual and predicted classes. The vertical axis represents the actual class of the input value. The horizontal axis represents the predicted class calculated by the classifier. The area percentages of a confusion matrix clarify how precise parts of the data set can be predicted. The evaluation reflects the already addressed good results but shows also some minor problems with the FIN-flood, SYN-ACK-flood and the Wi-Fi-deauthentication attack.

VII. AI-BASED MODULES

Artificial intelligence enables the consistent analysis of complex data through the use of special architectures and neural network techniques. In the following, the two architectures of the developed neural networks are presented. As shown in Figure 1, the AI-based modules are located in the third layer of the architecture. Two different models are developed. One is used to dectect anomalies through the use of an neural network, which is based on our previous publication where we described the theoretical approach. The other one is trained to classify attacks and is based on a pretrained VGG19 Convolutional Neural Network (CNN).

A. Anomaly Detection

To detect an attack there are two major ways, Signature Detection (SD) and Anomaly Detection (AD). The advantage of SD is, that known attacks can be detected very fast and with a high degree of precision. The downside is, that this method needs a well-maintained database with historical and actual attack signatures. This leads to a higher administrative burden and consequently, the system would be more costly. The AD avoids this disadvantage by monitoring the network and building a reference for the usual daily traffic. This allows the AD to recognize new and unknown attacks which would be overlooked by a SD-based system. But due to this characteristic, the AD is also prone to false-positive alarms because changes of the network traffic, for example, by bigger updates, can exceed the normal frame of reference.

1) Autoencoder-Anomaly-Detection-Model: To detect anomalies, we use a special neural network construction called Autoencoder. Their specific process logic allows the neural network to learn without any supervision. Autoencoder are useful tools for feature detection and can analyse unlabelled data with a large variety of different protocols. Autoencoder reduce a given input to the smaller dimensions. This has the consequence that the most important network information is elaborated. From this point on a reconstruction process is started to extrapolate the original input from the smallest reduced dimenson called bottelneck, as shown in Figure 6.

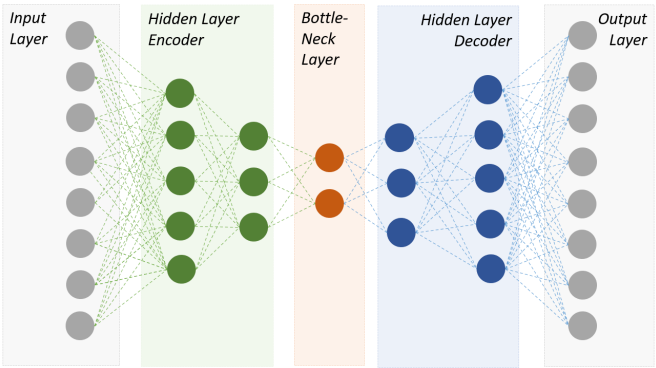


Figure 6. Basic Architecture of a Deep Autoencoder

After the training phase, the Autoencoder has learned to reconstruct the input information only based on the reduced information in the bottleneck. This means the reconstruction error of an extrapolated package compared to the input package is small. In reverse, the reconstruction of an attack package is not part of the trained behaviour of the neural network. Therefore the reconstruction error differs compared

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

to reconstruction error of normal network traffic. Based on the characteristics of the reconstruction error we can calculate the probability of an network anomaly. However, the Autoencoder cannot specify the specific kind of attack, this means classification models are good enhancements.

B. CNN-Classification-Model

Through a precise classification of threats, we gain additional information which can be used to deploy countermeasures in the reaction layer. The used classification model is based on the VGG19 Model, which was developed by the Visual Geometry Group of the University of Oxford [14].

1) CNN: The implementation of the classifier follows the assumption that the conversion of network packages into images can lead to better classification performance. To transform a packet into an RGB image all parameters had to be converted into a numerical value between 0 and 255. This range represents the 8 bits for each RGB colour channel. As described in Section VI-A all non-numeric values had to be transformed before rescaling. Based on this, a normalization with an Min-Max-Scaler was performed. Different to the procedure in Section VI-C the range for the Min-Max-Scaler was set to 0 – 255. To convert the rescaled data into an image, an array with 3072 elements was declared. The VGG19 algorithm requires 3 layers with a size of 32 by 32 pixels. Each layer represents the values of one RGB colour, i.e., red, green or blue. This leads to a bare minimum size of 3072 pixels for each image. Before the packet data is transferred, the array is initialised with 0, which can be seen as a representation of a fully black image. Thereafter the packet data is copied into the array and split into three parts, one for each layer. Every part is transformed into a 32 by 32 pixel layer and recombined with the other 2 layers to create a colour tensor that can be used by the VGG19 Model. First tests with this classification model delivered promising results. However, further tests with larger and more heterogeneous data sets are necessary to verify these results.

VIII. Conclusion and Future Work

The presented architecture provides the basis for the implementation of the static and AI-based methods for the iIDS. The conceptual design of iIDS is to combine different network monitoring methods in such a way that they can be operated both locally and in the cloud. The static methods are analyzing information of the recorded network traffic. In the same layer as the static analysis, EDA and data preparation are performed. The gathered information of EDA derives the most relevant features for the subsequent AI modules. Data preprocessing prepares the data set for the usage in static and AI modules. Different AI algorithms are implemented. The first module is used to detect anomalies using a special architecture of neural networks. By dimension reduction of the input space and subsequent extrapolation from the smaller dimension space, network anomalies can be detected by analyzing the reconstruction error expression. The second module is used for the classification of attacks. Here, data blocks are processed by time and number to create RBG image data. The convolutional neural network can then classify network attacks based on certain patterns within the image data. The data processing steps and data analysis methods described in the paper show that static and dynamic methods can be developed and combined in practice to provide better network monitoring.

In the future work, a more detailed evaluation layer is to be developed. In order to achieve the desired improvement, an algorithm will be developed to enhance the aggregation of the static and AI-based module results. Due to this changes the iIDS should be able to find even more appropriate countermeasures for detecting attacks.

References

[1] Robert Koch Institut, Demographischer Wandel, 30.07.2020, [online] Available at: https://www.rki.de/DE/Content/GesundAZ/D/Demographie_Wandel/Demographie_Wandel_node.html [retrieved: February, 2022]

[2] Ambient Assisted Living Deutschland - Technik die unser Leben vereinfacht, 2016 [online] Available at: http://www.aal-deutschland.de/ [retrieved: February, 2022]

[3] Bundesamt für Sicherheit in der Informationstechnik. 2022. Smart Meter Gateway Dreh- und Angelpunkt des intelligenten Messsystems. [Online] Available at: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Smart-metering/Smart-Meter-Gateway/smart-meter-gateway_node.html [retrieved: March, 2022].

[4] M. G. Avram, "Advantages and challenges of adopting cloud computing from an enterprise perspective", 2014, Procedia Technology Volume 12, p-529-534. [Online] Available at: https://www.sciencedirect.com/science/article/pii/S221201731300710X [retrieved: March, 2022].

[5] J. Graf, K. Neubauer, S. Fischer ,and R. Hackenberg, "Architecture of an intelligent Intrusion Detection System for Smart Home," 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2020, pp. 1-6

[6] R. Vinayakumar et al. "Deep Learning Approach for Intelligent Intrusion Detection System," in IEEE Access, vol. 7, pp. 41525-41550, 2019

[7] D. H. Summerville, K. M. Zach ,and Y. Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices," 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), 2015, pp.1-8

[8] C. D. McDermott, F. Majdani ,and A. V. Petrovski, "Botnet Detection in the Internet of Things using Deep Learning Approaches," 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-8,

[9] O. Brun et al. "Deep Learning with dense random neural network for detecting attacks against IoT-connected home environments", Communication in Computer and Information Science 821, vol.1, pp.79-89, February 2018

[10] M. Shah et al. "Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019, pp. 1-6.

[11] S. K. Mukhiya, U. Ahmed, Hands-On Exploratory Data Analysis with Python, Birmingham: Packt Publishing, 2020

[12] A. Burkov, The hundred-page machine learining book, by Andriy Burkov, Quebec City, Canada, 2019

[13] A. Ratner et al. Snorkel: rapid training data creation with weak supervision, Proc. VLDB Endow. 11, 3 (November 2017), pp.269–282.

[14] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," presented at 3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, pp.1-14.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

Cost-Effective Permanent Audit Trails for Securing SME Systems when Adopting Mobile Technologies

Bob Duncan
Department of Computing Science
University of Aberdeen
King's College, Aberdeen, UK
Email: robert.duncan@abdn.ac.uk and
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
Email: robert.duncan@arcada.fi

Magnus Westerlund
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
Email: magnus.westerlund@arcada.fi

***Abstract*—Cyber security for SMEs is a challenging activity. Since large corporations started to improve their cyber security process and strategies, this has made life considerably more challenging for attackers. This has resulted in a change of approach by attackers to pursuing SMEs. They have found such companies to be far less focussed on achieving really tight systems that are difficult to penetrate, to the extent that they would rather attack SMEs than large corporations. This is an important problem to deal with, because while large corporations who get successfully breached find the result expensive and time consuming to rectify, they usually have adequate reserves and resources to survive. This is seldom the case for SMEs, and up to 50% of successful breaches on SMEs can result in their bankruptcy. However, since SMEs have neither the reserves, resources nor sufficient skill levels of employees to help them deal with this difficult challenge, they are left at a considerable disadvantage. We propose a simple, economic approach that can improve security, ensure retention of a full forensic trail, all within their financial means.**

Index Terms*— *SMEs, mobile devices, cloud security, audit trails

I. INTRODUCTION

There is little doubt that keeping corporate systems secure presents a major challenge for businesses and in the UK, the Government Cyber Security Breaches Survey in 2020 [1] noted that almost half of all business suffered a breach during the previous year. While many large corporations have the necessary expertise and resources to deal with breaches, the same cannot be said for Small and Medium Sized Enterprises (SMEs). Why do we care about SMEs? Based on the World Trade Report, the International Federation of Accountants (IFAC) note that SMEs represent over 90% of the business population, 60-70% of employment and 55% of Gross Domestic Product (GDP) in all of the developed economies [2]. For SMEs, there are often serious constraints on resources, and limited expertise in this area among employees.

We have also seen a rapid evolution in business architecture, leading to new paradigms, such as cloud, distributed systems and so on. The evolving widespread shift to mobile communications and the ever increasing power of mobile phones, means that many employees will no longer operate just from a desk with a desktop computer, but instead might use a range of devices. Often any individual might have, in addition to the desktop, a laptop, a tablet, a mobile phone, or two, perhaps a smart watch to name but a few. These changes can offer improvements to the way business can operate more efficiently, but they also bring more risk. The traditional approach to security has been the "castle" approach to protect the centralised systems. Now add to this the effects of the pandemic and the move to working-from-home, often using completely insecure domestic network connections to add to the already precarious security approaches, and it is clear that a great many SMEs could be heading for disaster.

With limited expertise and resources and often limited understanding of the risks they face, this can put them at a serious competitive disadvantage. Perhaps far more worrying, will be the impact of their limited resources to spend on proper cyber security, leading to a continually increasing risk footprint. Another example might be that instead of using the largest Cloud Service Providers (CSPs) operated by 'big tech' companies such as Amazon Web Services, Microsoft Azure and Google Cloud, as well as traditional large corporations such as IBM and HP, many might be tempted to use smaller firms offering cheaper services, but who do not have the same security procedures in place as can be provided by the big CSPs.

Another important incentive concerns the ability to be compliant with ever more legislation and Regulation specifically targeting the proper control of Personally Identifiable Information (PII), such as we have seen with the introduction of the EU General Data Protection Regulation (GDPR). For a GDPR breach involving the PII of any EU resident, and UK residents, since it has been adopted by the UK since Brexit, fines can be levied at up to the greater of €20 million or 4% of annual turnover.

In Section II, we will consider the background on cyber security risks. In Section III, we will consider Cyber Security and SMEs and will introduce the framework for this interpretative study, where we will address Cyber Security Threats, SME behaviours, SME awareness and SME decision-making.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

In Section IV, we will consider some proposed corporate solutions, looking at how these might be overly complex and costly for SMEs to adopt. In Section V, we will consider how we could adapt these for SMEs in a more cost-effective and simple way to make them attractive to SMEs to implement. Finally, in Section VI, we will discuss our conclusion.

II. Background on Cyber Security Risks

Proper risk management for SMEs has become an ever more urgent and serious problem to address, often due to the availability of minimal resources and limited understanding of why this could be such a vital asset to the company [3]. All companies face a continuously increasing range of risk. While traditional disaster type risks are reasonably easy to understand, when it comes to data, the majority of these risks are adversarial risks, which can be much more difficult to predict, and thus identify properly. The big problem for SMEs, is that they stand a much higher chance of going bankrupt after a large breach, due to their much lower level of available resources [4].

Almost 40 years ago, Hollman and Mohammad-Zadeh [5] recognised the importance of proper risk management for SMEs, and 8 years later Miller [6] developed a very straightforward framework suitable for SMEs to use for this purpose. While the risks faced since that time have changed significantly, that is no excuse not to bother doing a good job of risk management, as the ultimate risk if nothing is done could lead to bankruptcy.

Falkner and Hiebl [7] suggested that SMEs generally faced 6 main categories of risk:

- Interest rate risk;
- Raw material price rise;
- E-business and technology risks;
- Supply chain risks;
- Growth risks;
- Management and employee risks.

The authors suggest that it is clear the area that SMEs need to focus on will be E-business and technology risks, and in particular, cyber security risk management. The vast majority of such risks faced are adversarial in nature, making them a much bigger challenge to deal with properly. Research in this area has been a bit on the sparse side, but back in 2003, Tranfield et al., [8], put some useful base information together in their paper that provides a good understanding to start from. It is very clear that the more SMEs can start to understand the true nature of the risks they face, the better they will be able to prepare themselves to defend against them.

III. Cyber Security and SMEs

Recently, Alahmari and Duncan [9] wrote about the challenges faced by SMEs and wrote about 5 areas of importance that ought to be considered:

- Cyber Security Threats in SMEs;
- Cyber Security Behaviours in SMEs;
- Cybersecurity Practices in SMEs;
- Cybersecurity Awareness in SMEs;
- Cyber Security Decision-Making in SMEs.

Thus we will consider their observations in each of the following 5 sub-sections:

A. Cyber Security Threats in SMEs

One of the key takeaways from this area is that a major challenge is to be able to articulate properly the concept of what exactly cyber security is and how it can impact on their business. Some of the key risks arise from cyber attacks that seek to breach data systems in order to steal, modify or delete data, or to make it inaccessible to the users of the business [10]. To this day, these risks continue to present the same level of challenges, other than that the frequency of such attacks has intensified during the past 7 years [11]. It is noticeable that attacks against SMEs have also increased during the pandemic.

On the risk assessment front, Barlette et al., [12], suggest it can be challenging for SMEs to be able to quantify exactly what the impact of breaches can be. The authors also suggest that while many SMEs believe they are not vulnerable because of their size. That is precisely why they are being attacked, since they present a much easier target than large corporations, due to their limited resources and understanding of what they face.

B. Cyber Security Behaviours in SMEs

Barlette et al., [12] also suggest that employee behaviour can expose SMEs to greater threats due to such practices as ignoring information policies, organizational guidelines, and company rules can lead to exposing the SME to much greater risk. Training and education are vital tools that can be used to improve user awareness. Of course, in some cases, [13], this might still not be enough, as they found in a previous study that while training uptake was 85%, the actual behaviour was much lower at 54%.

There is little doubt that user commitment and behaviour are vital elements that can play a significant role in the success of the business's ability to achieve a high level of security [14]. Indeed, Gundu [13] avers that the real problem is not employees knowledge and understanding, rather it is their general negative cyber security behaviour as a whole that is the cause.

C. Cyber Security Practices in SMEs

It has long been a concern in the literature that the SME approach to cyber security has been the lack of seriousness. SMEs have often failed to respond to the warnings coming from the cyber security community about cyber threat [15], and observations have often been made of how authorised people participate, albeit unconsciously, in risky practices which could have an adverse impact on the cyber security success of the SME.

Osborn and Simpson [16], suggest that most cyber security experts believe that current security practices used by SMEs could be a barrier to efficiency due to the lack of their engagement with the research community. The authors believe that is such a pity, since large corporations are benefitting

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

greatly from that relationship. While adopting outsourcing facilities such as the cloud could make a big contribution to the cyber security success of SMEs, their effectiveness would be seriously degraded by bad practices. As Bada et al., [17], suggest, failure to change those practices will perpetrate the continuing success of attackers leading to ever more attacks.

D. Cyber Security Awareness in SMEs

SME awareness of cyber security risks has traditionally been low, and Kaur and Mustafa, [14], suggest this has continually led to considerable risks to SME assets. Osborn and Simpson argue that the lack of knowledge of SME users significantly detracts from their awareness of cyber attacks, which leads to an adverse impact on achieving an adequate level of cyber security.

Osborn, [18], suggests that SMEs need additional information about possible vulnerabilities, rather more than they need the implementation of tools for evaluating self-assessed risks, meaning they should develop the content of their specific awareness programmes immediately. Gundu [13], suggests that creating the best possible awareness program could be far more productive for SMEs as a help to reduce the potential risks to an acceptable level.

The threat to cyber security has been recognised as the greatest threat to SMEs and that addressing this challenge by deploying protective measures alone is not enough. Kabanda et al., [15], suggest that increasing awareness is likely to have a far more positive impact.

E. Cyber Security Decision-Making in SMEs

In considering decision-making in information risk management, SMEs have adopted out-sourcing as part of their digital strategy. Successful out-sourcing has improved the web presence of SMEs and increased efficiency. However, outsourcing may have created a cyber security knowledge gap in SMEs. If cyber security is recognised, it is seen as a secondary issue to the presence. There is a lack of focus on security by design methodology in SMEs. Owners and managers generally play a major role [19] [12] [15], which demonstrates the importance that is understood at a managerial level. Such a pity that by the time it gets to implementation in SMEs that such poor results are achieved.

IV. Existing Corporate Solutions

In this section, we will take a look at a number of corporate solutions which have been developed to address a number of key areas to give a flavour of what large corporations can achieve with their vast reserves, resources and in-house expertise, in collaboration with the research community.

In 2016, Duncan and Whittington warned about forensic issues which they described as the Cloud Audit Problem [20], [21] and proposed a possible approach, although warning of some of the potential barriers faced. They followed this in 2017 with a suggestion on how to create such an immutable database and how to set it up to carry out such a task.

In 2018, Duncan and Zhao [22], considered the use of blockchain as an alternative to some of the conventional databases, which were limited in what they could do. At this time Neovius et al., [23] looked at the use of distributed ledger technology to provide much higher level security, and later adapted this approach to address IoT security weaknesses [24].

In 2019, Westerlund and Jaatun [25], addressed the challenge of dealing with the cloud forensic problem, while ensuring compliance can be achieved with the GDPR

In response to the serious weaknesses inherent in Internet of Things (IoT) devices, Wikström et al., [26] developed a high security approach using blockchain, but this time incorporating Ethereum smart contracts to extend the power of the work. This work would go on to be implemented to demonstrate the viability of the concept.

It would be important to recognise that these concepts were specifically targeted towards large corporations, meaning that they would have both sufficient resources available to develop and implement the full system and also would have sufficient in-house expertise to ensure proper configuration for the implementation would be carried out.

We must be clear that the required level of resource availability and in-house calibre of staff would likely be far in excess of anything that a great many SMEs would be able to provide. Thus we considered how we might come up with an effective, yet economical approach that could allow them to greatly improve their security capabilities.

V. Adapting Existing Corporate Solutions

The key requirement was therefore to keep it simple and find an approach that would be relatively straightforward to provide a much higher level of cyber security, without pushing them beyond their often constrained budgets. We also considered the fact that not all SMEs are equal. They might vary from a one man operation up to a large company size with many employees, much more resources than the smallest, and possibly more technically competent staff as well.

We decided to attempt to propose a basic level approach, then add incremental options depending on the resources and in-house expertise of the SME.

Even the smallest SME would likely operate with a considerable range of disparate devices, many of which would use different operating systems, different software and apps, which presents many SMEs with their first challenge. With insufficient resources and limited skills in the workforce, how could they do anything constructive from there?

Asking them to make changes to devices, or to update specialised software, would likely be a challenge too much for many. In many cases, there could also be a further issue, namely the Bring Your Own Device (BYOD) approach in many SMEs. Added to this would be the recent trend towards Working From Home (WFH) brought on by the Covid-19 pandemic.

A. The Basic Proposal

We felt the sensible thing would be to re-think how we might accomplish dealing with costly solutions for SMEs with

very limited resources. Since all devices used offsite, and with the BYOD use in the office, the sensible approach would be to set up a Virtual Private Network (VPN), require all sign-ins to corporate systems to be addressed through the VPN and capture the forensic records in one place. In this way, no mater whether company devices, BYOD devices on users working on site could all login to the VPN, which would provide better security and privacy and the relevant data collected from the VPN without any company users noticing any change in what they had to do, other than a new login method. In this way, all direct logins to the company systems could be blocked to enforce login via the VPN.

Many large providers can deliver a cloud-based VPN service for a very reasonable monthly cost. They can also provide systems geared for growing businesses. It is possible to rent a company dedicated server to ensure increased availability and having connecting up/down speeds of 1Gbps. These company based solutions offer a dashboard to monitor and control what is going on with the company network. An SME without any other provisioned public endpoints than a VPN, would be able to effectively block off external system access. Certain providers can also offer VPN access based on multi-factor authentication (MFA) that further strengthen the authentication by utilizing a secondary key. Hence, if user id and password combination leak, a secondary physical key will still hinder unauthorized VPN access.

We would need to add an immutable database, and a new open source offering came to market just over a year ago, called immudb, which offers the fastest of database capabilities without it being possible to tamper with the data contained within the database. This addresses all of the traditional issues with slow or unusable systems. The combination of the use of the VPN in conjunction with the immutabe database, not only addresses improving the security of access systems, it also offers to ensure complete forensic trails are maintained, all without the need for SMEs to to spend huge sums the simply do not have.

The immutable database can be be kept remote from the VPN server and could even be based on a cloud system. As long as security is tight for this server, then it will provide exactly what any SME would wish to have. A full forensic trail of every device attempting to access company systems. In this way, the company will be assured that only staff accessing company systems would be able to be granted access. All external users of company systems would effectively be blocked off completely, leading to much tighter security for the SME.

This setup could include ALL devices, including mobile phones, but some companies might prefer to have a more structured approach. We therefore look at how we might meet this need.

B. Adding Mobile Devices to the proposal

In the business world, the most popular mobile phone systems use either Android operating systems, as developed by Google, or Apple systems for a more up-market approach. Google, for example, offer a mobile desktop package that provides a management dashboard, known as Android Enterprise. This can be developed in conjunction with Google. The mobile is connected through the VPN, using a business profile configuration, thus allowing the corporation to monitor for threats and hinder access.

VI. Conclusion and Future Work

SMEs must realise that they can no longer afford to ignore the need to pay serious attention to detail in matters of security. They must also start to realise that in order to remain compliant with the range of legislative and regulatory compliance, there is a strong need to address cyber security risks head on.

Legislators and regulators will not accept any excuses when it comes to cyber breaches, especially where personally identifiable information is involved. There are no excuses, and the legislators and regulators are right to bring those companies who fail to keep users' data properly secured to account.

These proposals we have offered provide a minimum step on the route to proper security. This needs to be done properly, and we do recognise that many SMEs simply do not have the resources to achieve a robust level of security. These proposals offer a potential route to achieving a much improved level of security for an extremely modest cost. With the bare minimum of expense, an SME could begin the process of bringing their systems to a much more robust level.

It is fair to say that these proposals are designed as a first robust step, and there will be considerable improvements that can be made in future. This proposal allows for the easy add-on of additional protections, thus building on what would already be there.

We plan to test this approach in the near future to demonstrate how well this basic approach can work for SMEs. Once we confirm the effectiveness of the approach, we would look to develop the extra advances that would allow additional new features to be added.

References

[1] HMG, "UK Cyber Security Breaches Report 2020," HMG, London, Tech. Rep., 2020. [Online]. Available: https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2020 [Last Access: 25th February 2022]

[2] IFAC, "The Foundation for Economies Worldwide is small business." [Online]. Available: https://www.enisa.europa.eu/topics/national-cyber-security-strategies/sme_cybersecurity [Last Access: 25th February 2022]

[3] J. Brustbauer, "Enterprise risk management in SMEs: Towards a structural model," *International Small Business Journal*, vol. 34, no. 1, pp. 70–85, 2016.

[4] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi, "Decision support approaches for cyber security investment," *Decision support systems*, vol. 86, pp. 13–23, 2016.

[5] K. W. Hollman and S. Mohammad-Zadeh, "Risk management in small business," *J. Small Bus. Manag.*, vol. 1, pp. 47–55, 1984.

[6] K. D. Miller, "A framework for integrated risk management in international business," *Journal of international business studies*, vol. 23, no. 2, pp. 311–331, 1992.

[7] E. M. Falkner and M. R. W. Hiebl, "Risk management in SMEs: a systematic review of available evidence," *The Journal of Risk Finance*, 2015.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

[8] D. Tranfield, D. Denyer, and P. Smart, "Towards a methodology for developing evidence-informed management knowledge by means of systematic review," *British journal of management*, vol. 14, no. 3, pp. 207–222, 2003.

[9] A. Alahmari and B. Duncan, "Cybersecurity Risk Management in Small and Medium-Sized Enterprises: A Systematic Review of Recent Evidence," in *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. IEEE, 2020, pp. 1–5.

[10] K. Renaud and G. R. S. Weir, "Cybersecurity and the Unbearability of Uncertainty," in *2016 Cybersecurity and Cyberforensics Conference (CCC)*. IEEE, 2016, pp. 137–143.

[11] A. A. Alahmari and R. A. Duncan, "Investigating Potential Barriers to Cybersecurity Risk Management Investment in SMEs," in *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2021, pp. 1–6.

[12] Y. Barlette, K. Gundolf, and A. Jaouen, "CEOs' information security behavior in SMEs: Does ownership matter?" *Systemes d'information management*, vol. 22, no. 3, pp. 7–45, 2017.

[13] T. Gundu, "Acknowledging and reducing the knowing and doing gap in employee cybersecurity complaince," in *ICCWS 2019 14th International Conference on Cyber Warfare and Security*, 2019, pp. 94–102.

[14] J. Kaur and N. Mustafa, "Examining the effects of knowledge, attitude and behaviour on information security awareness: A case on SME," in *2013 International Conference on Research and Innovation in Information Systems (ICRIIS)*. IEEE, 2013, pp. 286–290.

[15] S. Kabanda, M. Tanner, and C. Kent, "Exploring SME cybersecurity practices in developing countries," *Journal of Organizational Computing and Electronic Commerce*, vol. 28, no. 3, pp. 269–282, 2018.

[16] E. Osborn and A. Simpson, "Risk and the Small-Scale Cyber Security Decision Making Dialogue—a UK Case Study," *The Computer Journal*, vol. 61, no. 4, pp. 472–495, 2018.

[17] M. Bada, A. M. Sasse, and J. R. C. Nurse, "Cyber security awareness campaigns: Why do they fail to change behaviour?" *arXiv preprint arXiv:1901.02672*, 2019.

[18] E. Osborn, "Business versus technology: Sources of the perceived lack of cyber security in SMEs," 2015.

[19] A. Bayaga, S. Flowerday, and L. Cilliers, "IT Risk and Chaos Theory: Effect on the performance of South African SMEs," in *WMSCI 2017 - 21st World Multi-Conference Syst. Cybern. Informatics, Proc.*, vol. 2, no. 5, 2017, pp. 48–53.

[20] B. Duncan and M. Whittington, "Enhancing Cloud Security and Privacy: The Cloud Audit Problem," in *Cloud Computing 2016: The Seventh International Conference on Cloud Computing, GRIDs, and Virtualization*. Rome: IEEE, 2016, pp. 119–124.

[21] B. Duncan and M. Whittington, "Enhancing Cloud Security and Privacy: The Power and the Weakness of the Audit Trail," in *Cloud Computing 2016: The Seventh International Conference on Cloud Computing, GRIDs, and Virtualization*, no. April. Rome: IEEE, 2016, pp. 125–130.

[22] Y. Zhao and B. Duncan, "Could Block Chain Technology Help Resolve the Cloud Forensic Problem?" in *Cloud Computing 2018: The Ninth International Conference on Cloud Computing, GRIDs, and Virtualization*, no. February. Barcelona, Spain: IARIA, 2018, pp. 39–44.

[23] M. Neovius, J. Karlsson, M. Westerlund, and G. Pulkkis, "Providing tamper-resistant audit trails for cloud forensics with distributed ledger based solutions," in *CLOUD COMPUTING 2018*, 2018, p. 29.

[24] M. Westerlund, M. Neovius, and G. Pulkkis, "Providing Tamper-Resistant Audit Trails with Distributed Ledger based Solutions for Forensics of IoT Systems using Cloud Resources," *International Journal on Advances in Security*, vol. 11, no. Number 3 & 4, pp. 223–231, 2018.

[25] M. Westerlund and M. G. Jaatun, "Tackling the cloud forensic problem while keeping your eye on the GDPR," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2019, pp. 418–423.

[26] J. Wikström, M. Westerlund, and G. Pulkkis, "Smart Contract based Distributed IoT Security: A Protocol for Autonomous Device Management," in *21st ACM/IEEE International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2021) (forthcoming)*, Melbourne, Australia, 2021.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

Towards Efficient Microservices Management Through Opportunistic Resource Reduction

Md Rajib Hossen
Dept of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX, USA
Email: mdrajib.hossen@mavs.uta.edu

Mohammad A. Islam
Dept of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX, USA
Email: mislam@uta.edu

***Abstract*—Cloud applications are moving towards microservice-based implementations where larger applications are broken into lighter-weight and loosely-coupled small services. Microservices offer significant benefits over monolithic applications as they are more easily deployable, highly scalable, and easy to update. However, resource management for microservices is challenging due to their number and complex interactions. Existing approaches either cannot capture the microservice inter-dependence or require extensive training data for their models and intentionally cause service level objective violations. In our work, we are developing a lightweight learning-based resource manager for microservices that does not require extensive data and avoid causing service level objective violation during learning. We start with ample resource allocation for microservices and identify resource reduction opportunities to gradually decrease the resource to efficient allocation. We demonstrate the main challenges in microservice resource allocation using three prototype applications and show preliminary results to support our design intuition.**

Index Terms*— *Microservices; Resource-Management; Cloud-Computing; Service-Level-Objective; Kubernetes.

I. Introduction

Cloud applications have been evolving from monolithic architecture to microservice architecture. For instance, leading IT companies, such as Netflix, Amazon, eBay, Spotify, and Uber are adopting microservices [1]–[3]. In contrast to monolithic applications with a few large layers [4], microservice architectural style consists of a set of loosely coupled small-scale services deployed independently, each with its process and database. The services communicate via lightweight communication mechanisms, such as HTTP API, gRpc [2], [5]–[8]. Figure 1 shows the difference between monolithic and microservice architectures. Compared to monolithic applications, microservices are more easily deployable, highly scalable, easy to update components, and have better fault tolerance.

Microservices, however, come with their own sets of challenges. As shown in Figure 1, microservices have complex communication between them. To complete a request, a microservice may call one or several other microservices in parallel or sequential order. Due to these complicated relationships, microservice resource management becomes challenging. Naive approaches may waste resources by over-provisioning or violate the Quality of Service (QoS) by underprovisioning. Moreover, current solutions for resource

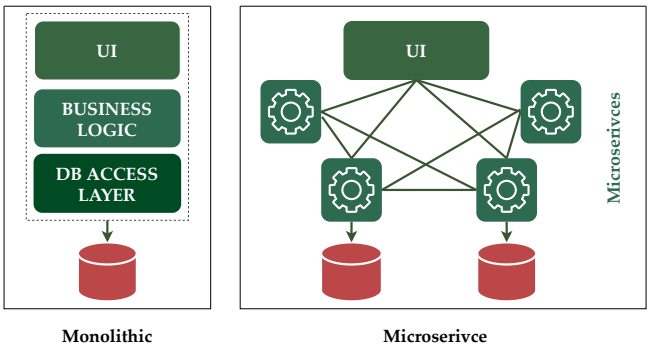


Fig. 1. Architecture of Monolithic and Microservices. Monothilic application blocks are defined and deployed as single units. Microservices offer more flexibility by decoupling an application into several small units and deploying independently.

management of clouds are for monolithic applications [9]–[14]. Although they provide excellent solutions for existing monolithic applications and data centers, they fail to capture the complicated relationship among microservices. As a result, these approaches can not be applied directly to microservices.

With the need for efficient resource management systems, research in resource management for microservices is gaining momentum in academia [15]–[21]. Prior works focusing on adopting popular approaches from monolithic applications such as allocation rules and model-based resource management fail to capture microservice inter-dependences and interactions in scalable fashion [15], [16]. Meanwhile, Machine Learning (ML) based approaches require extensive training data for building reliable models [17]–[19]. Their heavy dependency on data makes them slow to adapt to changes in microservice deployment such as software updates and hardware changes, even to changes in resource demand due to change workload intensity. More importantly, ML based approaches need to create Service Level Objective (SLO) violations to train the models intentionally [18], [20].

In this paper, we present our preliminary results towards developing a lighter-weight resource manager for microservice that learn efficient resource allocation through iterative interaction with the microservice application, yet do not rely on extensive data and avoid intentional SLO violation

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

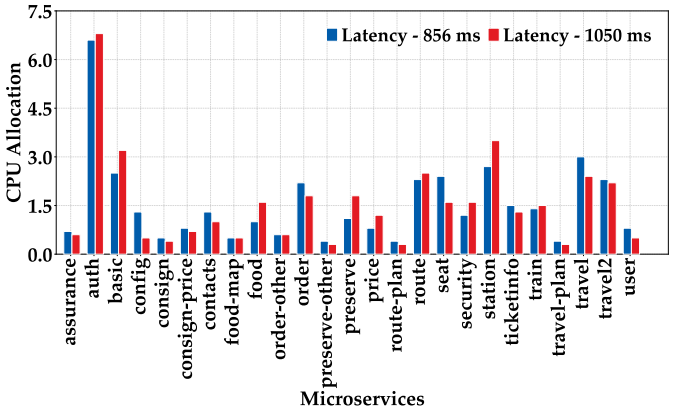
during the learning process. As a work-in-progress work, we show the key challenges of efficient resource allocation using three prototype microservice implementations. We then present our solution approach where we start with sufficient resource allocation for every microservice and then identify resource reduction opportunities to allocate efficiently. We avoid causing SLO violations during the learning since we always maintain at least more resources than required for each microservice. We also present experimental results from the prototype applications supporting our solution intuition. Finally, in future work, we discuss the technical challenges in our approach and our plans to address them.

The rest of the paper is organized as follows. We discuss related works in Section II. In Section III, we introduce the problem of microservice resource management and its challenges. We present our proposed solution in Section IV followed by concluding remarks and directions for future work in Section V.

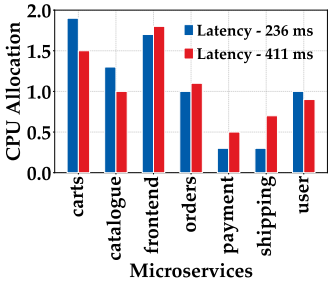
II. Related Work

Existing efforts on microservice resource management can be categorized as heuristics-based, model-based, and machine learning-based approaches. Kubernetes [22] is a popular container orchestration system that scales container resources horizontally and vertically. It decides resource allocation by defining thresholds for performance metrics such as CPU Utilization and Memory Utilization [23]. Kwan et al. [15] propose a rule-based solution to autoscale resources based on CPU and memory utilization. These rule-based systems require application profiling for identifying the threshold values that vary from application to application and require frequent updates with system changes. To ease these efforts, several other studies model-based resource management systems such as queue network, and application profiling based approach to find optimal resources [16].

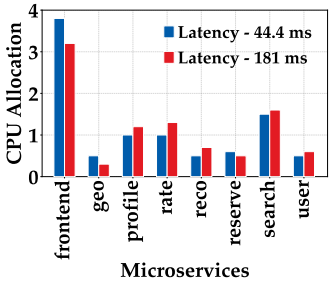
ML is another heavily used approach in resource management for microservices. ML approaches can be divided into 1) data-driven and 2) Reinforcement Learning approaches. Data-Driven approaches usually work by finding the relationship between performance metrics and response time [17]–[19]. Jindal et al. [17] collected data to calculate the serving capacity of microservices without violating SLO. Yu et al. [19] proposed an approach to identify scaling needed services by using the ratio of 50 percentile and 90 percentile response time and used Bayesian optimization to scale them horizontally. Zhang et al. [18] proposed a space exploration algorithm to gather data to train two machine learning models - Convolutional Neural Network and Boosted Trees. These models are then used to generate resources for various workloads. Although these papers consider microservice complexity when deciding resources, they need extensive experimental data and intentional SLO violations to train the models. For example, Zhang et al. [18] collected performance data after intentionally causing response time going 20% above the SLO. Such intentional SLO violations may not be feasible for production systems. Reinforcement Learning or Online Learning is used to manage resources dynamically for microservices [20], [21]. Qiu et al. [20] first identifies bottleneck microservices using a support vector machine and then uses reinforcement learning to provide resources for bottleneck microservices. However, they injected anomalies into the system to generate training data which may not be possible in real life. The AlphaR [21] uses a bipartite graph neural network to determine the application characteristics and then uses reinforcement learning to generate resource management policy. Although reinforcement learning provides online learning, they can suffers from a long training time.



(a) Train ticket (total CPU - 38.7)



(b) Sock Shop (total CPU - 7.5)



(c) Hotel reservation (total CPU - 9.4)

Fig. 2. Impact of resource distribution among microservice in response time. The X-axis shows the microservices in each applications, and the Y-axis shows the CPU allocation of each microservices.

III. Overview

In this Section, we formalize the problem statement, discuss the challenges in finding efficient resources, and describe the experimental settings for implementations.

A. Problem Statement

Using a discrete time-slotted model indexed by k where the resource management decisions are refreshed once every time slot, we formalize the microservice management as the following optimization problem, Efficient Microservice Management (EMM) where the objective is to minimize the

total resource allocation with the performance satisfying the SLO.

$$\text{EMM: minimize}_{\mathbf{r}(k)} \quad \sum_{n=1}^{N} r_n(k) \tag{1}$$

$$\text{subject to} \quad \mathcal{L}(\mathbf{r}(k)) \leq SLO, \tag{2}$$

Here, N is the number of microservices, $\mathbf{r}(k) = (r_1(k), r_2(k), \cdots, r_N(k))$ is the resource allocation, and $\mathcal{L}(\mathbf{r}(k))$ is the performance of the microservice application which is a function of the resource allocation vector $\mathbf{r}(k)$. We define the microservice performance (i.e., $\mathcal{L}(\mathbf{r}(k))$) as the end-to-end 95^{th} percentile response time.

B. Prototype Applications

To study EMM, we deploy three benchmark microservice applications - a ticket booking platform *Train Ticket* [24], a reservation application *Hotel Reservation* [8], and a e-commerce website *Sock Shop* [25]. The *Train Ticket*, *Sock Shop*, and *Hotel Reservation* applications consist of 41, 13, and 18 microservices, respectively. We deploy these applications on a Kubernetes [22] cluster with three nodes, each with two 20-cores Intel Xeon 4210R processors. In our resource allocation problem, we mainly consider CPU resource allocation for each microservice. As memory allocation cannot be easily changed without restarting the containers, we allocate enough memory to each container to ensure that the memory does not become the bottleneck resource. We can set the memory and the initial CPU allocation following offline profiling used in typical cloud applications [26]. Notably, in cloud deployments, it is a common practice to overprovision (e.g., allocate 20% more resource than required), which fits perfectly with our solution approach.. We also use Prometheus [27] and Linkerd [28] to collect performance metrics from containers and the applications. For all of our prototype implementations, we consider 95^{th} percentile end-to-end response time as the performance metrics.

C. Challenges in EMM

Solving EMM is particularly difficult for microservices because it is very hard to accurately estimate $\mathcal{L}(\mathbf{r}(k))$ in practice. The main reasons are that microservices are interdependent, and the behavior of microservices changes based on the workloads and CPU allocations. Moreover, each request to the applications needs to be processed by several microservices. Hence, all the microservices in the execution path dictate a request's end-to-end response time. If one microservice becomes a bottleneck, the end-to-end response time will increase. As a result, even with a fixed total CPU allocation, different CPU distributions (i.e., different $\mathbf{r}(k)$) among microservices may produce widely different response times. Figure 2 shows a motivating example, where we see that for the same amount of CPU allocation, the response time varies significantly when the resource distribution among microservices change.

In Figure 2(a), we see that the response time for train ticket increases more than 20% with an unfavorable allocation. In Figure 2(b), sock shop shows 74% increase and in Figure 2(c), hotel reservation shows more than 300% increase in response time due to change in resource distribution.

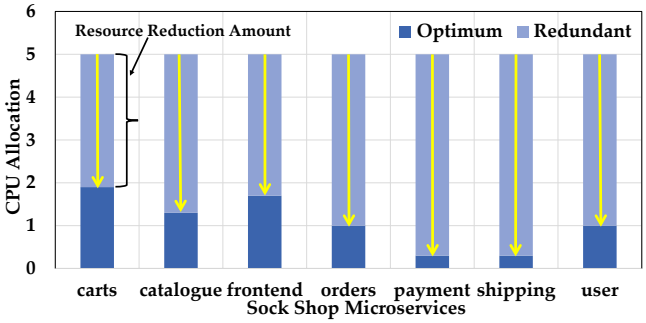


Fig. 3. Illustration of opportunistic resource reduction-based approach.

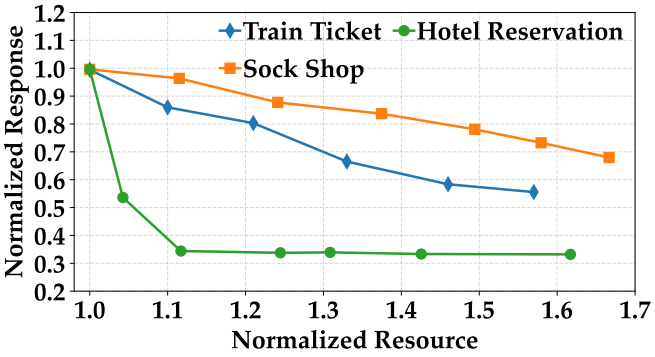


Fig. 4. Changes in response time with resource allocation demonstrating our intuition for opportunistic resource reduction-based approach.

IV. Opportunistic Resource Reduction

To circumvent the challenge of estimating the response time $\mathcal{L}(\mathbf{r}(k))$, we adopt a feedback-based approach where we do not need to know $\mathcal{L}(\mathbf{r}(k))$, instead we use the feedback from the application to find the response time for a given resource allocation. We collect this performance feedback at the end of a resource allocation time slot. We then can find the minimum resource allocation that satisfies the SLO by iteratively interacting (i.e., allocating resource and tracking its impact) with the application. This iterative interaction can be interpreted as a learning-based solution where we learn the efficient resource allocations.

However, since we are using a live system for the learning, we need to carefully decide our resource allocation in each iteration to avoid SLO violations. Hence, instead of finding efficient resource allocation, we reframe our solution approach to finding resource reduction opportunities. More specifically, we deploy our applications with sufficient resources to satisfy the SLO and then identify *opportunities for resource reduction* from different microservices based on the application's performance. Figure 3 illustrates our approach on sock shop application where we start with CPU allocation of "5" for each microservice and gradually reduce the resources in iterations

to reach the optimum allocation. Since we start from a high resource, no microservice becomes the bottleneck to cause SLO violation during this iterative resource reduction.

To corroborate the intuition of our approach, we run preliminary experiments on our microservice implementations. Our goal in these experiments is to show that a gradual decrease in the resource can lead to the optimum allocation. We first find the optimum resource allocation for each application by exhaustive trials and errors. We run each application several times, and during each new run, we increase the resource of a few randomly selected microservices. The response time of these experiments is shown in Figure 4 where we normalize each application's response time to their respective SLO and resource allocation to their respective optimum. We see that a gradual decrease in the resources leads the response time closer to the SLO. This results support our resource reduction-based design intuition. Moreover, it proves that, we can reach the optimum resource allocation eventually.

Note that, the trail and error approach is not applicable in a live system due to its impact on the QoS. Finally, in the above experiment that we randomly increase the resource instead of resource reduction as we have yet to develop our resource reduction algorithm. Nonetheless, the observation of the evolution of the response time with resource allocation changes in these experiments holds for the reduction algorithm.

V. Conclusion

This paper is part of our ongoing project of developing a complete resource manager that finds the optimum resources for every application without human intervention and without degrading the Quality of Services (QoS).

The preliminary results demonstrate the potential of opportunistic resource reduction. To identify the resource reduction opportunity, we plan to use the difference between the response time and the SLO. This is because, in general, resource reduction increases in response time. Hence, any room for response time increase (i.e., the difference between response time and SLO) can be interpreted as a resource reduction opportunity. The response time is an application-level metric and does not reveal which microservices are the best candidates for resource reduction. Hence, we plan to incorporate microservice-level performance metrics in our resource reduction algorithm. We will follow two principles to avoid SLO violations for future iterations. First, we will maintain microservice-level performance (e.g., utilization) upper limits for SLO compliance. We will refrain from further resource reduction if current metrics exceed the upper limit. The limits will be dynamically updated based on SLO satisfaction. Second, we will be conservative in resource reduction. Instead of reducing a considerable amount of resources in one iteration, we will use a gradient descent approach to reduce a small number of resources and then examine the system performances. This way, it will be guaranteed that even if the system can not find exact optimum resources, it will not violate SLO. In addition to avoiding SLO violations, we will also incorporate workload changes to the optimum resource allocation as the workload intensity directly affects the resource requirement for satisfying SLO.

References

[1] "Mastering chaos: A netflix guide to microservices," https://www.infoq.com/presentations/netflix-chaos-microservices/, accessed: 01/07/2022.

[2] "Introduction to microservices," https://www.nginx.com/blog/introduction-to-microservices, accessed: 01/05/2022.

[3] "Leading companies embracing microservices," https://www.divante.com/blog/10-companies-that-implemented-the-microservice-architecture-and-paved-the-way-for-others, accessed: 01/08/2022.

[4] "The definition of monolithic," https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/, accessed: 01/05/2022.

[5] "The definition of microservice," https://martinfowler.com/microservices/, accessed: 01/05/2022.

[6] "What are microservices?" https://microservices.io/, accessed: 01/05/2022.

[7] "Microservice vs monolithic architectures," https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/, accessed: 01/05/2022.

[8] Y. G. et al., "An open-source benchmark suite for microservices and their hardware-software implications for cloud and edge systems," in *International Conference on ASPLOS*, 2019, p. 3–18.

[9] C. Qu, R. N. Calheiros, and R. Buyya, "Auto-scaling web applications in clouds: A taxonomy and survey," in *ACM Comput. Surv.*, vol. 51, no. 4, Jul. 2018, pp. 1–33.

[10] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair scheduling for distributed computing clusters," in *ACM SIGOPS 22nd SOSP*, 2009, p. 261–276.

[11] "Hadoop fair scheduler," https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/FairScheduler.html, last Accessed: 01/05/2022.

[12] "Amazon aws autoscale," https://docs.aws.amazon.com/autoscaling/index.html, last Accessed: 10/05/2021.

[13] "Azure autoscale," https://azure.microsoft.com/en-us/features/autoscale/, last Accessed: 10/05/2021.

[14] M. Wajahat, A. A. Karve, A. Kochut, and A. Gandhi, "Mlscale: A machine learning based application-agnostic autoscaler," *Sustain. Comput. Informatics Syst.*, vol. 22, pp. 287–299, 2019.

[15] A. Kwan, J. Wong, H.-A. Jacobsen, and V. Muthusamy, "Hyscale: Hybrid and network scaling of dockerized microservices in cloud data centres," in *2019 IEEE 39th ICDCS*, 2019, pp. 80–90.

[16] A. U. Gias, G. Casale, and M. Woodside, "Atom: Model-driven autoscaling for microservices," *Proceedings - International Conference on Distributed Computing Systems*, vol. 2019-July, pp. 1994–2004, 2019.

[17] A. Jindal, V. Podolskiy, and M. Gerndt, "Performance modeling for cloud microservice applications," in *ACM/SPEC ICPE*, 2019, p. 25–32.

[18] Y. Zhang, W. Hua, Z. Zhou, G. E. Suh, and C. Delimitrou, "Sinan: Ml-based and qos-aware resource management for cloud microservices," in *International Conference on ASPLOS*, 2021, pp. 167–181.

[19] G. Yu, P. Chen, and Z. Zheng, "Microscaler: Automatic scaling for microservices with an online learning approach," *ICWS 2019 - Part of the 2019 IEEE World Congress on Services*, pp. 68–75, 2019.

[20] H. Qiu, S. S. Banerjee, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer, "FIRM: An intelligent fine-grained resource management framework for slo-oriented microservices," in *14th USENIX Symposium on OSDI*, Nov. 2020, pp. 805–825.

[21] X. H. et al., "Alphar: Learning-powered resource management for irregular, dynamic microservice graph," in *2021 IEEE IPDPS*, 2021, pp. 797–806.

[22] "Kubernetes: Production grade container orchestration," https://kubernetes.io/, accessed: 01/20/2022.

[23] "Kubernetes horizontal pod autoscaler," https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/, last Accessed: 10/05/2021.

[24] Z. et al., "Fault analysis and debugging of microservice systems: Industrial survey, benchmark system, and empirical study," *IEEE Transactions on Software Engineering*, vol. 47, no. 2, pp. 243–260, 2021.

[25] "Sock shop microservice demo," https://microservices-demo.github.io/, accessed: 08/31/2021.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

[26] Y. Gan, M. Liang, S. Dev, D. Lo, and C. Delimitrou, "Sage: Practical and scalable ml-driven performance debugging in microservices," in *26th ACM International Conference on ASPLOS*, 2021, p. 135–151.
[27] "Prometheus - from metrics to insights," https://prometheus.io/, last Accessed: 10/08/2021.
[28] "Linkerd: A different kind of service mesh," https://linkerd.io/, accessed: 08/31/2021.

Copyright (c) IARIA, 2022. ISBN: 978-1-61208-948-5

Gestalt Computing: Hybrid Traditional HPC and Cloud Hardware and Software Support

Jay Lofstead
Sandia National Laboratories
Albuquerque, NM, USA
email: gflofst@sandia.gov

Andrew Younge
Sandia National Laboratories
Albuquerque, NM, USA
email: ajyoung@sandia.gov

***Abstract*—Traditional modeling and simulation-based HPC workloads demand a platform optimized for low latency node-to-node communication and a write intensive IO load. Cloud-based applications tend to not need the fast communication and are frequently heavily read dependent. Both of these workloads are immensely important and demand larger and larger machines to handle the demand. Unfortunately, budgets demand that large scale compute be a shared resource for both workloads in spite of this opposed design priority. Through the use of persistent memory (PMEM) devices on node, dynamically we can reconfigure the nodes to either support fast reading through using the on node-PMEM as a reading cache or as slow DRAM to reduce the time spent communicating with logical neighbor nodes. We outline a vision for a machine hardware and software architecture dynamically shifting to simultaneously support both workloads as demand dictates with minimal additional cost and complexity.**

***Index Terms*—HPC, Cloud, hybrid computing, hybrid workloads, persistent memory, job scheduling**

I. Introduction

Modern large scale computing is no longer just used for uniform modeling and simulation (modsim) workloads, but also incorporates data intensive workloads like machine learning and other data analytics techniques. The tools on both sides have matured considerably and both offer excellent support for their respective workloads. The new challenge is the realization that traditional HPC workload data is being processed by these newer data analytic techniques. The arriving challenge is that the generated modsim data needs to be processed as it is generated eliminating the need to store raw data.

Traditional HPC platforms are organized to best support *large-scale, scale-up* workloads. These are a single task that work on a single or a small number of very large data items iteratively to explore physics phenomena in some way. These very large data items can be 1 PB or larger demanding spreading the computation across many nodes not just for processing speed, but also simply to be able to hold the data in working memory.

These scale-up platforms use high performance, low latency interconnect networks, such as InfiniBand [1] and Slingshot [2], to reduce the communication overheads of the frequent data exchange operations. Each large data item represents something, such as a volume of air in a room. To look at air flow patterns, the individual air molecules, or some representative value, must be tracked on a short distance periodic basis in the entire volume. The aggregate size can be extremely large if extended to something like the global atmosphere. To evolve the computation, some force in the form of air molecules with directional velocity are inserted pushing the existing molecules forming air currents. To perform the calculations, the data is split into sub-volumes, each assigned to a compute unit of some sort, and calculated independently. Since the force is not confined to any given sub-volume, after each calculation round, the edges are exchanged with the logical neighbors to enable more independent computation. Periodically, all of the data is written to persistent storage for offline analysis of the simulation evolution. One challenge being faced today and continuing to worsen is that the storage IO bandwidth is not growing fast enough to absorb data at the rate scientists are willing to pay for. In essence, unless the scientists want to spend the vast majority of their compute time allocation performing IO to storage, they need to write far less often than they would prefer. Instead, they want to perform their data analysis and processing tasks, which can reduce data volumes dramatically, as part of their computation process.

The new generation HPC workloads run on platforms organized to best support *large-scale, scale-out* workloads. These are many tasks that are used to process a very large number of relatively tiny data items and use parallelism to accelerate the computation. Additional compute nodes are used to increase compute speed, but each computation is largely independent, allowing seemingly unlimited scaling with linear speed increases simply by adding more compute capacity.

This new generation, as it encompasses such a large number of new markets and types of data processing, has prompted developing rich, accessible tools that with a little work, can also process the scale-up data. The quality of these tools has prompted scientists to demand support for incorporating them directly into their workflows with no regard to the underlying hardware and software system support. System administrators are left trying to best support user demands, but with the wrong tools.

The underlying software infrastructure required for these scale-out tools generally fits better in an orchestration system. Current examples such as Kubernetes [3], OpenStack [4], and Docker Swarm [5] are optimized to dynamically start and stop a number of service instances on an as demanded basis.

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

Resource sharing performance penalties are a lower concern since processing is largely independent.

Further motivating this situation are budgetary concerns from funding agencies. High-end machines are very expensive to field and operate and the thought of fielding two machines of similar scale, but a bit different design just to support two different workloads is not seen as a responsible use of budget. Instead, the funding agencies have been prompting the laboratories to find ways to make these workloads co-exist on a single, slightly larger platform. This will be slightly more expensive than one to both field and operate and still be able to handle the aggregate computational needs.

While cloud bursting [6] and multi-cloud deployments may be able to address the needs in theory, other policy or security concerns may demand a fully secured, on site deployment. Export controlled, sensitive, or classified data or processing have special requirements of the cloud platform and the connection with it. Certification [7] can allow workloads and data for some while higher consequence and more sensitive information still cannot use these infrastructures. This prompts a need for hybrid on-site resources that can easily handle both the traditional modsim workloads as well as the data analytics workloads.

The cost of provisioning a system to fully support both workloads is prohibitive. Excess DRAM capacity would go unused for many workloads while node-local high capacity, low cost storage (e.g., disk or equivalent) introduces even more costs and failure points. This excess is financially impossible in essentially all cases. Instead, there needs to be a technology that can serve both to expand DRAM capacity as well as serve as node-local storage. Persistent Memory (PMEM) offers such a solution.

PMEM devices live on the memory bus and are accessed via load/store instructions. At the same time, compared to DRAM prices, they are cheaper for capacity. This makes them ideal for expanding DRAM capacity at a reasonable cost. PMEM devices, because they offer the additional persistence property, also offer a way to store larger data quantities on node with less concern about failures. This dual property makes PMEM an intriguing solution to the hybrid device need. Unfortunately, hardware alone is not sufficient to realize the gestalt platform ideal.

This paper explores a potential systems and software architecture leveraging persistent memory devices, such as the Intel Optane [8], that could offer higher memory capacity for heavily compute intensive workloads and near-compute storage for data throughput-intensive workloads.

The rest of this paper is structured as follows. First, in Section II, we discuss persistent memory devices, their characteristics, and how we propose they can be used. Section III, we describe the software environment required to support these hybrid workloads. Two use cases are presented demonstrating the potential of this design in Section IV. Next in Section V, we discuss some related work. Finally, Section VI offers takeaway requirements and challenges we need to address as a community to achieve a true gestalt platform capable of easily supporting any mix of traditional HPC mod-sim and data analytics workloads simultaneously.

II. Persistent Memory

Persistent memory (PMEM) devices offer higher capacity at lower performance and cost than traditional DRAM devices. One commercially available option is the Intel Optane [8]. Unfortunately, confusing naming hides these devices among different technology suppressing marketplace visibility. In spite of this, considerable work [9]–[16] has been done exploring the potential for use as both extended memory (volatile-style) and storage (persistent-style) devices. This hybrid nature offers a dual use technology that can be the key enabling hardware for a gestalt workload.

Typically, to extend memory into a persistent storage media requires going through a special interface, such as SATA [17] or the PCIe [18] bus based NVMe [19] devices. Fast, solid state devices can be deployed through these interfaces, but interacting with them is fundamentally different from accessing regular memory. Instead of a typical load/store instruction, working through a device driver of some sort is required. The access granularity is typically a block, which may range from 4 KB to even 1 MB. Reading a single byte from these devices causes accessing an entire block at a time. The block is then managed in the device driver's memory and the single byte is returned to the caller.

While systems like `mmap` [20] can hide a block-storage device behind a memory-like standard API, all it does is to translate the memory accesses into block-storage accesses with a memory page granularity. Systems like DAX [21] can more directly map devices for access, but it is specialized hardware for accessing slower devices than PMEM.

Prior to PMEM devices, NVMe devices held the fastest storage devices title. While NVMe is relatively expensive compared to spinning media (i.e., disks), the performance is dramatically different making it a difficult choice to deploy disks except for large capacity, slow access devices. NVMe devices are not without their limitations. Ceph [22] rewrote their low-level device access code because their disk and SSD (over SATA) optimized code was too slow to enable streaming to NVMe devices at hardware rates. Achieving the full performance potential is possible, but takes some work.

Compare that configuration against how PMEM devices are deployed. PMEM devices are placed on the memory bus and are accessed via load/store instructions within the CPU. The access granularity is a cache line. Compatible CPUs have been modified to tolerate the longer latencies involved with using PMEM devices to avoid timeouts or other failure conditions. As far as the machine code is concerned, these are normal main memory devices with no special software drivers or other interfaces necessary. The performance characteristics are 3x-5x faster than an NVMe device.

Alternatively, technologies like CXL [23] offer a way to place these devices across the interconnect for shared use. Through a CXL accessible pool, a large scale, shared, persistent store can be configured for the machine simplifying and

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

accelerating data sharing among nodes. A recent announcement that Pacific Northwest National Laboratory and Micron are partnering to investigate this setup to accelerate HPC with ML/AI workloads [24].

Standard IO routines are tuned for standard IO interfaces making NVMe devices easy to deploy. A little software tuning and significant performance advantages can be realized. PMEM devices require specialized interface code to achieve the full potential. The pMEMCPY [16] library demonstrates that without special tuning, even a highly optimized IO library can only achieve about half the available performance of a PMEM device. The extra copies in the OS kernel must be avoided to get the full performance benefits.

A. Machine Architecture

Our proposed hardware architecture is illustrated in Figure 1. It consists of a CPU and GPU both with high bandwidth memory and additional PMEM devices capable of holding as much as 3x the CPU and/or GPU memory. This 3x capacity has been used for several DOE Leadership Computing capacity calculations for burst buffers (e.g., Summit at ORNL). Adopting this standard makes reasonable sense based on this standard. This enables compute to swap to the PMEM devices for significantly longer independent computation phases, the ability to write data for slower migration off node, and to hold a significant corpus for data analytics tasks to avoid going off node to load data.

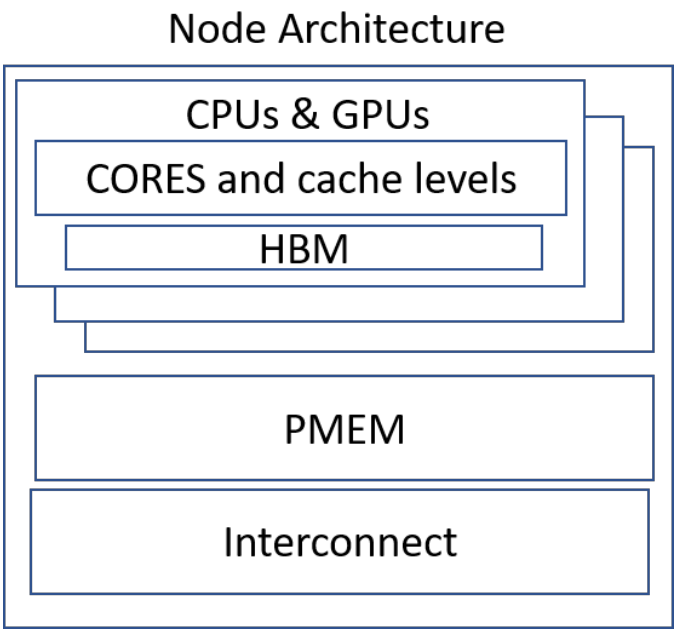


Fig. 1. Proposed Node-level architecture

The interconnect would need to be a traditional HPC-style interconnect optimized for low latency, high bandwidth operation, such as InfiniBand. Storage would be in a large, shared pool that would serve as scale-up scratch for offloading from the compute nodes or for staging into the compute area prior to running an analytics job.

Otherwise, the machine architecture could be relatively "standard". For example, the scratch/shared storage pool could be a large scale distributed or parallel storage system with various tiers from scratch to campaign storage to archive. This part of the architecture is typically very similar for cloud or HPC deployments. The main differentiator would likely be in a cloud deployment, there is more redundancy with assumed failures while HPC would focus on more reliability and ability to recover from failures. With standard components used across both types of machines, the costs are essentially the same with the software layer differentiating the operational aspects.

B. Discussion

PMEM devices can fit into the niche needed to take largely similar platforms and enable them to dynamically adapt to whatever workload is deployed on them. With an adequate software layer on top, this gestalt can be achieved.

Complicating this picture are the pricing differences. PMEM can achieve around 3x the performance of NVMe devices, but cost 5x as much. Unless that performance gain is essential and tuning the IO libraries to fully take advantage of PMEM is possible, it is unlikely worth the extra cost except in specialized cases. For the purposes of this paper, we are considering two cases. The first is where the cost is less important than the raw achievable performance for time sensitive applications. The second is the potential for eliminating an entire platform's cost by deploying this more expensive technology into an existing platform and adapting the software infrastructure to form the gestalt.

III. Software

Software infrastructure has a long way to go. The cloud native structure of using a minimal virtual machine to host containers for the functionality is the right start. For this system design, we could configure for deployment a minimum of two Virtual Machines (VMs) or a spectrum with different ratios of memory-configured or storage-configured PMEM devices. HPC system administrators are starting to understand the benefit of this approach, but are still concerned about performance loss. Given the financial realities, accepting a tiny performance loss to support diverse workloads may be the price paid.

While the cloud-native software infrastructure may seem to be adequate, it does not support bulk-synchronous parallel workloads well. Cloud-like infrastructure focuses on spinning up or down independent service instances, potentially all of the same type. Scale-up HPC best supports a single large instance spread across potentially all of the nodes. Startup times should be very similar to avoid wasted compute times as no process can really get started until all of them have started. The same is true for the synchronization points. All task processes communicate to exchange data after each computation phase. While alternative programming models, such as Charm++ [25] and Legion [26] seek to break this dependency, they instead rely on oversubscription to cores and delayed computation

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

to handle communication overheads and delays. That way computation is never waiting for data to arrive. Net compute time may reduce, but wall clock time may increase, but on fewer resources. Further, re-architecting codes into this model is considerable, non-trivial effort. Finally, the compatibility of this scale-up model with cloud software infrastructures is unproven.

Using containers for software deployment makes sense for many reasons. First, immutable, stored containers offer a precise reference to what code was run for future publications and reproducibility. Second, containers eliminate missing or conflicting third party library problems. As long as the container itself can work, then it can work in any compatible container hosting environment. Third, Continuous Integration/Continuous Deployment infrastructure is often designed to work with containers to perform the regular automated testing tasks. We propose using containers for all of the software elements given the natural deployment capabilities developed in the cloud space in addition to others. A simple picture of this software architecture is presented in Figure 2.

Ideally, all data would be in containers as well. A prototype of this model called Data Pallets [27] later updated in Olaya's Master's thesis [28], shows the potential for enhancing not just reproduciblity, but also traceability and understandability for workflows.

Immutable rather than ephemeral containers are a key element. By preserving, unchanged, the containers, reproducibility is easier to verify. Associating container hash values with the resulting output offers a strong connection to how the data is created.

This reproducibility benefit can be enhanced by storing the virtual machine images used, along with the container hash codes, to offer a stronger reproducibility foundation for research. Long term, hardware emulators would be necessary to replace obsolete/unavailable CPUs or other infrastructure, but that is a solvable problem.

Finally, combining all of these together requires a system such as a fully realized Fuzzball [29], to manage deployment. Better effort on simplifying the system and improving performance is needed. Application startup time should be at most seconds rather than several minutes, as it stands in the prototype version available today. Fuzzball seeks to support both traditional scale-up and scale-out workloads simultaneously with a ground-up re-build of the support infrastructure. This kind of rethinking of the software layer is essential to a hybrid platform and it requires PMEM, or some similar concept, to be able to contain costs while meeting the true gestalt.

IV. Use Cases

The first use case is a traditional modsim workload that couples with data analytics code on separate, or even the same, node(s) to form a single workflow. The second use case is a traditional cloud data analytics workflow. A third is a traditional modsim alone.

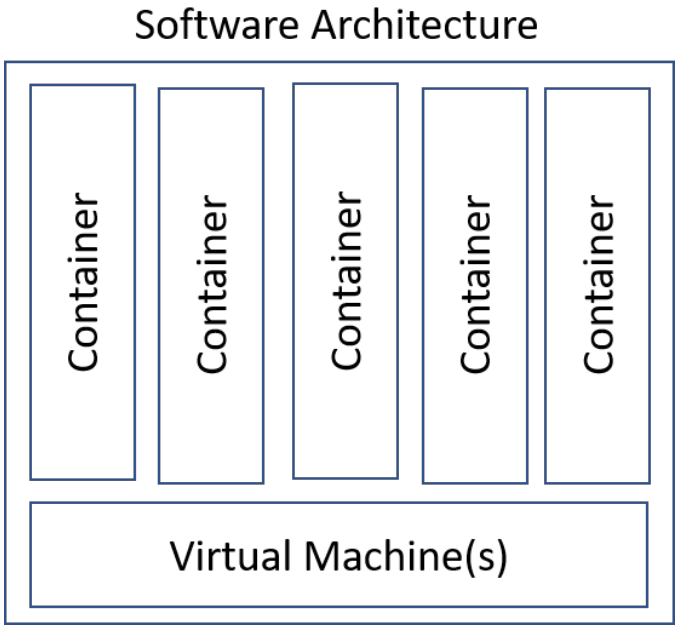


Fig. 2. Proposed software architecture

A. Use Case 1: Modsim + Data Analytics

Consider a case with a large scale physics simulation, such as LAMMPS [30]. LAMMPS is a molecular dynamics simulator that shows the behavior of molecules or atoms under various conditions. Depending on the physics complexity, the amount of data per process can vary widely. For this use case, assume the physics is moderately complex and LAMMPS is configured to run on a few dozen nodes. For example, a fracture simulator or a material melting would both be appropriate. In both cases, the material volume would be split across many processes, but the interaction with neighboring atoms and molecules affects what happens with each other. On several additional nodes, analysis code evaluates the data to determine the presence or absence of a fracture, determined by inter-atom distances, or does a visualization enabling storing just the image rather than the raw data.

LAMMPS would run normally pushing data to the analysis nodes periodically. With moderate physics, LAMMPS can use more on-node memory swapping from HBM to the PMEM to reduce the communication overheads. For the data analysis/visualization processes, they need the data to do their analysis and rendering making storage a better choice. For many of these routines, they are already written assuming they will read from storage rather than memory. Reconfiguring the PMEM to be storage devices enables these codes to run against PMEM as storage without modification.

B. Use Case 2: Cloud Data Analytics

For a large scale machine learning model generator, reading and re-reading LOTS of different data items can be necessary. To avoid contention in both the interconnect and on storage, it is best to pull the data down to the nodes once and then read

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

and re-read as appropriate. In this case, deploying all of the PMEM as storage devices makes the most sense.

C. Use Case 3: Traditional Modsim

Using the LAMMPS example again from above, consider if LAMMPS is running alone and just needs to push output to storage periodically. In this case, the PMEM could be configured partially as memory and partially as storage to serve as a staging area (burst buffer) that is sent asynchronously to off-node storage during the computation intensive phase.

D. Discussion

For all cases, the underlying hardware can be adapted to best suit the needs of the particular software involved. In the first case, the simulation code uses the PMEM as on-node fast swap, but pushes output to the PMEM on the nodes with data analytics code for further processing and ultimately permanent storage. In the second case, the PMEM is all allocated for read-ahead buffers accelerating the processing. The third case balances use between two needs enabling both more memory and node-local fast storage. These cases show the potential of the hardware/software design.

V. Related Work

Related work focuses on scheduling systems and some specialized platforms. First the various kinds of scheduling systems are briefly discussed and then the platforms are examined. This examined more in depth in previous work [6]. Task scheduling system generally fall into one of three categories with some newer systems attempting to bridge either the categories, platforms supported, or both.

A. Grid Scheduling

Grid systems predate clouds and were limited by deploying on system specific hardware and software. Cloud abstracted much of that away making it easier to move code from one system to another. In spite of that, the grid era has technology that is still relevant and useful today.

Globus [31] offers a tools suite to handle much of the data movement needs, but it does not offer the mature compute differentiation needed by these platforms.

To handle hybrid workloads, cross-platform grid scheduling was attempted [32]. This work, and all similar work, all suffered from complexity of different back-end system features and architectures, among other limitations. In short, it was unable to expose the desired functionality at a complexity level acceptable to users.

B. Scale Up Task Scheduling

Traditional HPC, scale-up workloads use long standard schedulers, such as Slurm [33]. As an open source tool with strong community support, it has grown to handle most HPC compute management needs. Slurm has grown new features, such as multi-resource scheduling [34], but that is different than what the gestalt is trying to do. For multi-resource scheduling, it assumes that a resource has a single purpose and just needs to be scheduled along with another resource (e.g., burst buffer capacity along with compute notes). We are proposing a different model with uniform hardware that is dynamically configured based on the workload needs. This eliminates the need for the more complex multi-resource scheduling.

A next generation HPC scheduler, Flux [35] is trying to offer a hybrid HPC and cloud scheduling capability. However, Flux is just starting to explore how to incorporate cloud resources and has only scratched the surface of the vast complexities involved. It will take considerable time and effort for Flux to successfully integrate a single cloud platform. Much like Slurm, this is focused on placing workloads on uniform platforms strictly optimized for one workload type or the other. Data movement issues are a daunting problem they have not addressed.

C. Scale Out Task Scheduling

The scale-out schedulers focus on large numbers of small tasks that often run quickly and exit. Scheduling throughput of 1000s of tasks per second is needed to keep the machine busy. Orchestration systems, discussed later, are a variation on this approach with some different constraints.

The need to handle scheduling a large number of short running tasks prompted the creation of Sparrow [36] and similar systems. This shift from the traditional very large single task with a long run time demanded these new tools to better handle resource use.

Spark [37] is another popular system that generally scales well. Mesos [38] with systems like Aurora [39] and Yarn [40] offer examples of high throughput task oriented schedulers.

Other systems like Omega [41] were built in frustration of the need to support heterogeneous clusters that evolved as new hardware was added over time with broken and obsolete hardware decommissioned. The priority for a system like this is to support a wide variety of hardware features and enable a reasonably efficient mapping from job requirements onto the available hardware balancing needs against availability. This is different from our goal of uniform hardware, but varying software.

Some modern software engineering architectures, such as function-as-a-service [42], embrace this kind of short task execution model as a central feature.

D. Container Orchestration

The alternative approach for job scheduling has shifted more to container management rather than task management. Systems like Kubernetes [3] and Docker Swarm [5] offer increasingly rich and complex environments for deploying long-lived services that can dynamically scale on demand. This architecture has compatibility issues with traditional HPC workloads, such as deployment time and potentially resource sharing. For service orchestration, all instances being fully deployed at nearly the same time is not as important as all of them running eventually, but soon. That makes this model less attractive to those wanting more efficient machine usage.

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

E. Hybrid Schedulers

New systems, like Fuzzball [29], intend to work like Flux, but are rethinking the environment from the ground up. While the ideas are inspiring, these systems need considerable work before they can achieve the goals outlined above. For example, the software system configuration for Fuzzball requires a full Kubernetes system install to manage the scale out workload and it does not address the very large scale data items in terms of moving compute to different kinds of resources. While these are planned to be addressed eventually, the fully needed functionality is still an unsolved and unimplemented problem.

F. Other Related Work

The literature that discusses HPC systems at national laboratories [43]–[45] provides rich information about HPC usage trends, resource utilization metrics, evolution of supercomputers over time, among many other user- and system-centered topics. Most of the existing studies on HPC environments—both historic and also recent—pay little attention to cloud computing and its benefits, considering integration with clouds to be secondary or optional in nature. With the shifting workload demands, revisiting these investigations is an important priority.

Among the counterexamples, a study of computing resources at the Texas Advanced Computing Center (TACC) [46] stands out as it describes a considerable number of cloud-style jobs being processed as a result of integration of TACC facilities with Jetstream [47], a cloud computing facility sponsored and managed by the USA's National Science Foundation (NSF).

Chameleon Cloud [48] offers a bare-metal-as-a-service cloud option. While this is an NSF supported effort focused on supporting both research and education, the resources are not as extreme as leadership computing facilities. Instead, the focus is on supporting smaller scale efforts with a strong tie to educational environments rather than production-style workloads. The bare metal aspect is quite appealing for our model as the full software stack is deployed at runtime on a per job basis. Shifting this to be more dynamic to a per node or per core basis would make this a fine contender for supporting the gestalt we propose.

An additional advantage of this approach concerns power usage [49]. By using PMEM devices, the energy footprint can be reduced. This also applies to ML applications [50].

The final part of the related work concerns cloud bursting. These systems look at how to use a cloud resource as "overflow" for an onsite or just another large scale compute resource. Work on these bursting approaches [51]–[53] show the challenges and potential for making these systems work. Microsoft, with the Azure platform, offer this as a fundamental part of their cloud strategy. They encourage users to install an Azure instance at their premises and to use Microsoft's private cloud instances as bursting capacity. This enables customers to right size their on site compute resources to control costs while not hitting limits for transient peak workloads. Achieving this kind of balance for HPC and Cloud would offer an excellent balance by deploying jobs that do not need the HPC platform characteristics onto the associated cloud when there is demand for the HPC platform specific characteristics by jobs. However, these capabilities still do not exist. Further, the most daunting problem of moving large data items from one platform to another is left completely unaddressed.

G. Discussion

The kinds of workloads each of these system classes addresses is different and difficult to address with a single scheduler and resource management system. This has led to the fragmentation of platform development efforts, where each platform is essentially treated as an independent direction for research and development and optimized to best address its own, particular subset of the workloads.

The need for a hybrid use platform as we describe exists today. With machine learning tools being incorporated into scientific simulations, both worlds must co-exist simultaneously on the same platform or latency will dominate the computations. For climate simulations [54], [55], machine learning models are substituting for parts of the model that may have too many parameters or the physics is not fully understood. Using models generated from observational data, reasonable estimates of these effects improve the simulation model quality overall. Using ML to perform initial analytics is also growing in popularity.

VI. Conclusions

Overall, we propose that using persistent memory devices, along with appropriate system software that can dynamically on a node-by-node, job-by-job basis make a single platform capable of efficiently handling both traditional modsim scale-up workloads coupled with data analytics scale out workloads. While persistent memory devices are currently cost prohibitive, NVMe devices offer a more affordable, and still relatively performant option that can work similarly with a little software help. We propose that this architecture be adopted for future systems.

Acknowledgements

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] G. F. Pfister, "An introduction to the infiniband architecture," *High performance mass storage and parallel I/O*, vol. 42, no. 617-632, p. 102, 2001.

[2] D. De Sensi, S. Di Girolamo, K. H. McMahon, D. Roweth, and T. Hoefler, "An in-depth analysis of the slingshot interconnect," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–14, IEEE, 2020.

[3] E. A. Brewer, "Kubernetes and the path to cloud native," in *Proceedings of the sixth ACM symposium on cloud computing*, pp. 167–167, 2015.

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

[4] T. Rosado and J. Bernardino, "An overview of openstack architecture," in *Proceedings of the 18th International Database Engineering & Applications Symposium*, IDEAS '14, (New York, NY, USA), p. 366–367, Association for Computing Machinery, 2014.

[5] J. Turnbull, *The Docker Book: Containerization is the new virtualization*. James Turnbull, 2014.

[6] J. Lofstead and D. Duplyakin, "Take me to the clouds above: Bridging on site hpc with clouds for capacity workloads," *CLOUD COMPUTING 2021*, p. 63, 2021.

[7] D. I. S. Agency, "Department of defense cloud computing security requirements guide." https://rmf.org/wp-content/uploads/2018/05/Cloud_Computing_SRG_v1r3.pdf, 2017.

[8] O. Patil, L. Ionkov, J. Lee, F. Mueller, and M. Lang, "Performance characterization of a dram-nvm hybrid memory architecture for hpc applications using intel optane dc persistent memory modules," in *Proceedings of the International Symposium on Memory Systems*, MEMSYS '19, (New York, NY, USA), p. 288–303, Association for Computing Machinery, 2019.

[9] D. Li, J. S. Vetter, G. Marin, C. McCurdy, C. Cira, Z. Liu, and W. Yu, "Identifying opportunities for byte-addressable non-volatile memory in extreme-scale scientific applications," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium*, pp. 945–956, IEEE, 2012.

[10] S. Mittal and J. S. Vetter, "A survey of software techniques for using non-volatile memories for storage and main memory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1537–1550, 2015.

[11] S. R. Dulloor, S. Kumar, A. Keshavamurthy, P. Lantz, D. Reddy, R. Sankaran, and J. Jackson, "System software for persistent memory," in *Proceedings of the Ninth European Conference on Computer Systems*, pp. 1–15, 2014.

[12] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee, "Better i/o through byte-addressable, persistent memory," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 133–146, 2009.

[13] J. Yang, J. Kim, M. Hoseinzadeh, J. Izraelevitz, and S. Swanson, "An empirical guide to the behavior and use of scalable persistent memory," in *18th USENIX Conference on File and Storage Technologies (FAST 20)*, pp. 169–182, 2020.

[14] Y. Shan, S.-Y. Tsai, and Y. Zhang, "Distributed shared persistent memory," in *Proceedings of the 2017 Symposium on Cloud Computing*, pp. 323–337, 2017.

[15] A. Kalia, D. Andersen, and M. Kaminsky, "Challenges and solutions for fast remote persistent memory access," in *Proceedings of the 11th ACM Symposium on Cloud Computing*, pp. 105–119, 2020.

[16] L. Logan, J. Lofstead, S. Levy, P. Widener, X.-H. Sun, and A. Kougkas, "pmemcpy: a simple, lightweight, and portable i/o library for storing data in persistent memory," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 664–670, 2021.

[17] K. Grimsrud and H. Smith, *Serial ATA Storage Architecture and Applications: Designing High-Performance, Cost-Effective I/O Solutions*. Intel press, 2003.

[18] D. Mayhew and V. Krishnan, "Pci express and advanced switching: evolutionary path to building next generation interconnects," in *11th Symposium on High Performance Interconnects, 2003. Proceedings.*, pp. 21–29, 2003.

[19] T. Coughlin, "Evolving storage technology in consumer electronic products [the art of storage]," *IEEE Consumer Electronics Magazine*, vol. 2, no. 2, pp. 59–63, 2013.

[20] L.-x. Wang and J. Kang, "Mmap system transfer in linux virtual memory management," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 1, pp. 675–679, IEEE, 2009.

[21] L. K. Foundation, "Linux Kernel Documentation: Direct Access for Files." https://www.kernel.org/doc/Documentation/filesystems/dax.txt, 2014.

[22] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proceedings of the 7th symposium on Operating systems design and implementation*, pp. 307–320, 2006.

[23] S. Van Doren, "Hoti 2019: Compute express link," in *2019 IEEE Symposium on High-Performance Interconnects (HOTI)*, pp. 18–18, IEEE, 2019.

[24] J. Russell, "PNNL, Micron Work on New Memory Architecture for Blended HPC/AI Workflows." https://www.hpcwire.com/2022/03/09/pnnl-micron-work-on-new-memory-architecture-for-blended-hpc-ai-workflows/, March 2022.

[25] L. V. Kale and S. Krishnan, "Charm++ a portable concurrent object oriented system based on c++," in *Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications*, pp. 91–108, 1993.

[26] M. E. Bauer, *Legion: Programming distributed heterogeneous architectures with logical regions*. Stanford University, 2014.

[27] J. Lofstead, J. Baker, and A. Younge, "Data pallets: containerizing storage for reproducibility and traceability," in *International Conference on High Performance Computing*, pp. 36–45, Springer, 2019.

[28] P. Olaya, J. Lofstead, and M. Taufer, "Building containerized environments for reproducibility and traceability of scientific workflows," *arXiv preprint arXiv:2009.08495*, 2020.

[29] CIQ, "Fuzzball: HPC-2.0." https://ciq.co/fuzzball/, 2022.

[30] S. Plimpton, P. Crozier, and A. Thompson, "Lammps-large-scale atomic/molecular massively parallel simulator," *Sandia National Laboratories*, vol. 18, p. 43, 2007.

[31] I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, no. 2, pp. 115–128, 1997.

[32] D. M. Batista, N. L. S. da Fonseca, and F. K. Miyazawa, "A set of schedulers for grid networks," in *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, (New York, NY, USA), p. 209–213, Association for Computing Machinery, 2007.

[33] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Workshop on job scheduling strategies for parallel processing*, pp. 44–60, Springer, 2003.

[34] Y. Fan, Z. Lan, P. Rich, W. E. Allcock, M. E. Papka, B. Austin, and D. Paul, "Scheduling beyond cpus for hpc," in *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*, pp. 97–108, 2019.

[35] D. H. Ahn, J. Garlick, M. Grondona, D. Lipari, B. Springmeyer, and M. Schulz, "Flux: A next-generation resource management framework for large hpc centers," in *2014 43rd International Conference on Parallel Processing Workshops*, pp. 9–17, IEEE, 2014.

[36] K. Ousterhout, P. Wendell, M. Zaharia, and I. Stoica, "Sparrow: distributed, low latency scheduling," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pp. 69–84, 2013.

[37] D. Cheng, Y. Chen, X. Zhou, D. Gmach, and D. Milojicic, "Adaptive scheduling of parallel jobs in spark streaming," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[38] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. H. Katz, S. Shenker, and I. Stoica, "Mesos: A platform for fine-grained resource sharing in the data center.," in *NSDI*, vol. 11, pp. 22–22, 2011.

[39] F. Pfeiffer, "A scalable and resilient microservice environment with apache mesos and apache aurora," *SREcon15 Europe*, May 2015.

[40] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, *et al.*, "Apache hadoop yarn: Yet another resource negotiator," in *Proceedings of the 4th annual Symposium on Cloud Computing*, pp. 1–16, 2013.

[41] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes, "Omega: flexible, scalable schedulers for large compute clusters," in *Proceedings of the 8th ACM European Conference on Computer Systems*, pp. 351–364, 2013.

[42] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, "A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms," in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 162–169, 2017.

[43] T. Patel, Z. Liu, R. Kettimuthu, P. Rich, W. Allcock, and D. Tiwari, "Job characteristics on large-scale systems: Long-term analysis, quantification and implications," in *2020 SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1186–1202, IEEE Computer Society, 2020.

[44] G. P. Rodrigo, P.-O. Östberg, E. Elmroth, K. Antypas, R. Gerber, and L. Ramakrishnan, "Towards understanding hpc users and systems: a nersc case study," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 206–221, 2018.

[45] G. Amvrosiadis, J. W. Park, G. R. Ganger, G. A. Gibson, E. Baseman, and N. DeBardeleben, "On the diversity of cluster workloads and its impact on research results," in *2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)*, pp. 533–546, 2018.

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5

[46] N. A. Simakov, J. P. White, R. L. DeLeon, S. M. Gallo, M. D. Jones, J. T. Palmer, B. Plessinger, and T. R. Furlani, "A workload analysis of nsf's innovative hpc resources using xdmod," *arXiv preprint arXiv:1801.04306*, 2018.

[47] C. A. Stewart, T. M. Cockerill, I. Foster, D. Hancock, N. Merchant, E. Skidmore, D. Stanzione, J. Taylor, S. Tuecke, G. Turner, *et al.*, "Jetstream: a self-provisioned, scalable science and engineering cloud environment," in *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, pp. 1–8, 2015.

[48] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock, *et al.*, "Lessons learned from the chameleon testbed," in *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, pp. 219–233, 2020.

[49] R. Farber, "BOOSTING MEMORY CAPACITY AND PERFORMANCE WHILE SAVING MEGAWATTS." https://www.nextplatform.com/2020/12/08/boosting-memory-capacity-and-performance-while-saving-megawatts/, 2020.

[50] N. Hemsoth, "STORAGE PIONEER ON WHAT THE FUTURE HOLDS FOR IN-MEMORY AI." https://www.nextplatform.com/2021/01/18/storage-pioneer-on-what-the-future-holds-for-in-memory-ai/, 1 2021.

[51] T. Bicer, D. Chiu, and G. Agrawal, "A framework for data-intensive computing with cloud bursting," in *2011 IEEE international conference on cluster computing*, pp. 169–177, IEEE, 2011.

[52] A. Gupta, P. Faraboschi, F. Gioachin, L. V. Kale, R. Kaufmann, B.-S. Lee, V. March, D. Milojicic, and C. H. Suen, "Evaluating and improving the performance and scheduling of hpc applications in cloud," *IEEE Transactions on Cloud Computing*, vol. 4, no. 3, pp. 307–321, 2014.

[53] W. C. Proctor, M. Packard, A. Jamthe, R. Cardone, and J. Stubbs, "Virtualizing the stampede2 supercomputer with applications to hpc in the cloud," in *Proceedings of the Practice and Experience on Advanced Research Computing*, pp. 1–6, ACM, 2018.

[54] P. A. O'Gorman and J. G. Dwyer, "Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events," *Journal of Advances in Modeling Earth Systems*, vol. 10, no. 10, pp. 2548–2563, 2018.

[55] V. M. Krasnopolsky and M. S. Fox-Rabinovitz, "Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction," *Neural Networks*, vol. 19, no. 2, pp. 122–134, 2006.

Copyright (c) The Government of Sandia National Laboratories, 2022. Used by permission to IARIA. ISBN: 978-1-61208-948-5