UMEÅ UNIVERSITY

Autonomous Resource Management for Mobile Edge Clouds

Chanh Nguyen

Licentiate Thesis, September 2019
Department of Computing Science
Umeå University
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

chanh@cs.umu.se

ISBN 978-91-7855-116-3
ISSN 0348-0542
UMINF 19.07

Electronic version available at http://umu.diva-portal.org/
Printed by UmU Print Service, Umeå University
Umeå, Sweden 2019

Abstract

Mobile Edge Clouds (MECs) are platforms that complement today's centralized clouds by distributing computing and storage capacity across the edge of the network, in Edge Data Centers (EDCs) located in close proximity to end-users. They are particularly attractive because of their potential benefits for the delivery of bandwidth-hungry, latency-critical applications. However, the control of resource allocation and provisioning in MECs is challenging because of the heterogeneous distributed resource capacity of EDCs as well as the need for flexibility in application deployment and the dynamic nature of mobile users. To realize the potential of MECs, efficient resource management systems that can deal with these challenges must be designed and built.

This thesis focuses on two problems. The first relates to the fact that it is unrealistic to expect MECs to become successful based solely on MEC-native applications. Thus, to spur the development of MECs, we *investigated the benefits MECs can offer to non-MEC-native applications*, i.e., applications not specifically engineered for MECs. One class of popular applications that may benefit strongly from deployment on MECs are cloud-native applications, particularly microservice-based applications with high deployment flexibility. We therefore quantified the performance of cloud-native applications deployed using resources from both cloud datacenters and edge locations. We also developed a network communication profiling tool to identify the aspects of these applications that reduce the benefits they derive from deployment on MECs, and proposed design improvements that would allow such applications to better exploit MECs' capabilities.

The second problem examined in this thesis relates to the dynamic nature of resource demand in MECs. To overcome the challenges arising from this dynamicity, we make use of statistical time series models and machine learning techniques to develop two *workload prediction models* for EDCs that account for both user mobility and the correlation of workload changes among EDCs in close physical proximity.

Preface

This thesis contains a brief introduction to Mobile Edge Clouds (MECs) infrastructures, a discussion on the challenges and problems to resource management in MECs, and the following papers[†]:

Paper I **Chanh Nguyen**, Amardeep Mehta, Cristian Klein, and Erik Elmroth. Why Cloud Applications Are not Ready for the Edge (yet). *4th ACM/IEEE Symposium on Edge Computing (SEC 2019)*, to appear, 2019.

Paper II **Chanh Nguyen**, Cristian Klein, and Erik Elmroth. Location-aware load prediction in Edge Data Centers. *2nd IEEE International Conference on on Fog and Mobile Edge Computing (FMEC 2017)*, pp. 25-31, IEEE, 2017.

Paper III **Chanh Nguyen**, Cristian Klein, and Erik Elmroth. Multivariate Long Short-term Memory based Location-aware load prediction in Edge Data Centers. *19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (IEEE/ACM CCGrid 2019)*, pp. 341-350, IEEE, 2019.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

[†]The included articles have been reformatted to comply with the thesis layout.

Acknowledgements

First and foremost, I would like to express my deep gratitude to my supervisor Prof. Erik Elmroth for giving me the great opportunity to pursue my studies in his group, as well as all his continuous support, guidance, and encouragement on the way.

My sincere thanks also goes to my co-supervisor, Cristian Klein, for his never-ending enthusiasm and generous help on developing my research skills, providing insightful comments, suggestions on my research, as well as proof-reading the thesis. Cristian is truly a mentor who always motivating me to do my best!

I would like to thank current and former colleagues at the Distributed Systems Research group: Abel, Ahmed, Amardeep, Danlani, Ewnetu, Gonzalo, Jakub, Johan, Lars, Luis, Mina, Monowar, Muyi, Petter, P–O, Selome, Thang, and Tobias – for contributing to such a friendly working atmosphere.

I would like to thank the colleagues from the Department of Computing Science at Umeå University for their support, sharing time together during lunches, fikas, and other enjoyable events.

This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. WASP does not only contribute with funding, but also an invaluable network with people from all over the world. Thank you all WASP seniors for arranging courses, conferences and international study trips, and to all the WASP batch 1 students for the fun moments we shared together!

On a personal level, I would like to give heartfelt thanks to my parents, my brother, and my parents-in-law for always being there for me with their endless support, and love.

Lastly, I thank my wife Trang, and my son, Nhat Minh for standing beside me, especially my wife for her endurance, support, and unwavering love that will always be in my heart. These few words are not enough to express my deepest appreciation for the efforts she has done during past years.

Umeå, September 2019
Chanh Nguyen

Contents

Chapter 1

Introduction

1.1 Background and Research Motivation

Thanks to remarkable advances in Artificial Intelligence (AI), Internet of Things (IoTs), and mobile technologies, recent years have seen a wave of disruptive applications and services in domains as varied as health care [1], industrial process control [2], intelligent transportation [3], and entertainment [4]. Unlike traditional cloud applications, these emerging applications are extremely latency-sensitive, produce massive quantities of data that is only of local interest, and require significant data processing capabilities as well as privacy guarantees.

Fifth generation (5G) wireless networks promise to deliver high performance in the form of extremely high bandwidth and ultra-robust low latency, with round-trip latencies below 1 ms [5]. These disruptive capabilities make it likely that 5G will be a perfect tool for overcoming many barriers to the success of emerging applications. However, there is an intrinsic problem with the current application deployment mechanism in which application-hosting nodes (i.e., centralized data centers) are located at a large distance from the end-users. As shown in Figure 1.1, the network traffic associated with cloud applications may have to traverse three different network layers:

- The *last-mile*: the link between the end-user and the edge network of an Internet Service Provider.
- The *aggregation*: the link between the edge network and the point at which the Internet Service Provider hands off the aggregated traffic to various network points of another provider.
- The *core network*: where off-premises or cloud data centers are situated.

Technically, congestion in the *last-mile* can be mitigated by deploying new broadband and radio access technologies such as 5G that increase the available

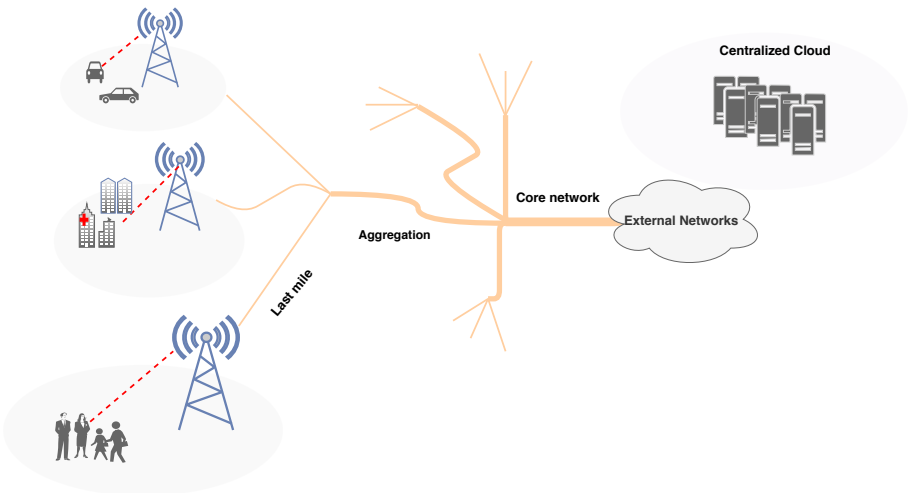


Figure 1.1: Congestion in the core network.

bandwidth. However, congestion can easily occur in the *aggregation* and *core network* if the aggregated demand exceeds the bandwidth available in these parts of the network. Because of this problem, modern telecoms networks are expected to cope poorly with the enormous and rapidly varying capacity demands that will arise in the near future.

For example, large-scale and industrial IoT systems such as those used in smart cities or oil pipeline monitoring. These systems typically feature millions of network-connected sensors that generate vast streams of data to support online analytics and real-time decision-making. As predicted by Ericsson and Cisco [6, 7], by 2021 there will be around 28 billion devices connected to the Internet (including 16 billion IoT devices), generating almost 850 zettabytes of data. Most of data generated by IoT devices is local in scope. That is to say, it is used for local purposes such as coordinating the movements of self-driving cars at a specific traffic hotspot, evaluating gas transmission pipeline state information, or exercising intelligent control over industrial processes in smart factories. It is extremely costly to transmit the large amounts of data generated by these devices to a centralized datacenter where data processing services are deployed. Additionally, transmitting such data can easily cause congestion in the aggregation and core networks if the aggregated demand exceeds the network capacity, and network latency and jitter can hurt latency-sensitive applications, resulting in potential damage to people and the environment.

High response times also cause significant performance problems for state of the art human-computer applications such as Virtual and Augmented Reality (VR, AR) applications and interactive online games that use computing resources located in distant datacenters. A compelling AR system must sup-

port High-Dynamic Range to ensure that the appearance of its virtual objects is spatially and temporally consistent with the real world. This requires a latency of no more than 10 ms [8]. Recent measurements indicate that typical network latencies between end users and public cloud datacenters are at least 20–40 ms over high-quality wired networks, and on the order of 100–250ms over a 4G network connection [9]. These latencies are too high for VR and AR applications to deliver instantaneous responses that appear natural to end-users because delivering such responses requires large volumes of data to be rapidly processed using complex technologies such as 3D rendering and machine vision.

There has recently been an explosion of edge content services such as YouTube Live, Facebook Live, and video surveillance that generate large quantities of high definition video data (e.g., live streams of sporting events). Current technologies that use a centralized datacenter architecture to process and deliver content of this type to millions of users are inefficient for three main reasons. First, forwarding such large quantities of data to the centralized datacenter places huge pressure on the core network. Second, in some cases, the aggregated latency between the distant datacenter and the end-users can cause a poor quality of service. Third, there is a high risk that transmitting such data will violate local network privacy policies due to a lack of location awareness and data privacy protection [10].

Modern cloud computing platforms have been widely used over the last decade because of their ability to offer computing services on demand at a low cost [11]. Many organizations have exploited the benefits they offer in terms of IT efficiency and business agility by deploying or migrating various applications to cloud platforms, including web servers, data processing tools, and batch job applications. However, cloud resources are centralized in a relatively small number of large datacenters located far from end-users, which implies an increase in network latency and jitter [12]. These issues are the Achilles' heel of cloud platforms, and make them unable to meet key requirements for the successful wide-scale use of mobile and IoT applications. Accordingly, studies on the impact of cloud deployment on the performance of IoT applications have shown that current compute clouds are sub-optimal for such applications because of the high latency and poor connectivity caused by long-distance communication [13, 14].

To address these challenges, current cloud computing systems must undergo a paradigm shift aimed at reducing overall network latency and jitter, minimizing the central cloud's ingress bandwidth requirements, and increasing reliability and flexibility in deploying and removing network functions in response to users' demands. As a result, recent years have seen a transition towards a new type of computing infrastructure called Mobile Edge Clouds (MECs) in which resource capabilities are distributed at the edge of the network, in close proximity to end-users. This wide geographical distribution of resources allows MECs to complement existing large-scale cloud platforms, making it possible to perform computation and data processing both at centralized datacenters and at the network edge. Computation and processing at the network edge

is achieved by exploiting the compute capacity of small servers or datacenters known as Edge Data Centers (EDCs) that are attached to radio base stations. Several concepts similar to MECs have been proposed, including Cloudlets [15], Fog Computing [16], Follow me Cloud [17], and Telco Cloud [18].

1.2 Characteristics of Mobile Edge Clouds

Figure 1.2 depicts an MEC system in which EDCs with heterogeneous scales and costs are distributed in close proximity to end-users in a wireless access network. This enable MECs to provide computation and storage capabilities with higher bandwidth and lower latency than would be possible for a centralized cloud. MECs also offer users other attractive benefits, such as the ability to run locally-targeted, context-aware services on EDCs that are closely-coupled to the radio network. This is particularly valuable for services that require guaranteed robust or low-latency communication, send a lot of data from end-user devices, or require analysis of enormous amounts of data immediately after its capture. It also allows network operators to provide additional value-added services and improve the experience of end users while alleviating security and privacy concerns. MECs have the following key characteristics:

Ultra low latency. Because their resources are physically close to end users, MECs can take advantage of 5G networks to achieve extremely low latency (on the order of several milliseconds).

Highly distributed and heterogeneous resources. The Edge Data Centers of MECs are distributed in different geographical locations and at different hierarchical levels within the wireless access network (i.e., at cellular base stations and access points). Further, unlike centralized datacenters, EDCs vary in scale and in terms of their processing and storage resources as well as their level of network connectivity and bandwidth.

Local network status awareness and local user context awareness. Since MECs' resources are deployed at the edge of the network, they can access real-time wireless network and channel information. Applications deployed on MECs can thus leverage location and user context data to provide a better service that is more accurately targeted to the end-user's circumstances (e.g., traffic assistance applications can give more accurate and helpful traffic information at a hotspot to specific end-users close to that hotspot).

Support for mobility. End-users typically access MECs via mobile devices and often change their points of attachment to the network. Therefore, mobility support is critical for MECs.

Interplay with central clouds. MECs complement traditional central clouds. Because their resources are distributed in the vicinity of the end-users, MECs can provide localized processing with context awareness and low latency. Conversely, more distant centralized clouds have much greater computing and storage capabilities, while being less costly than MECs because they are located in more sparsely-populated areas with access to cheap electricity and cooling.

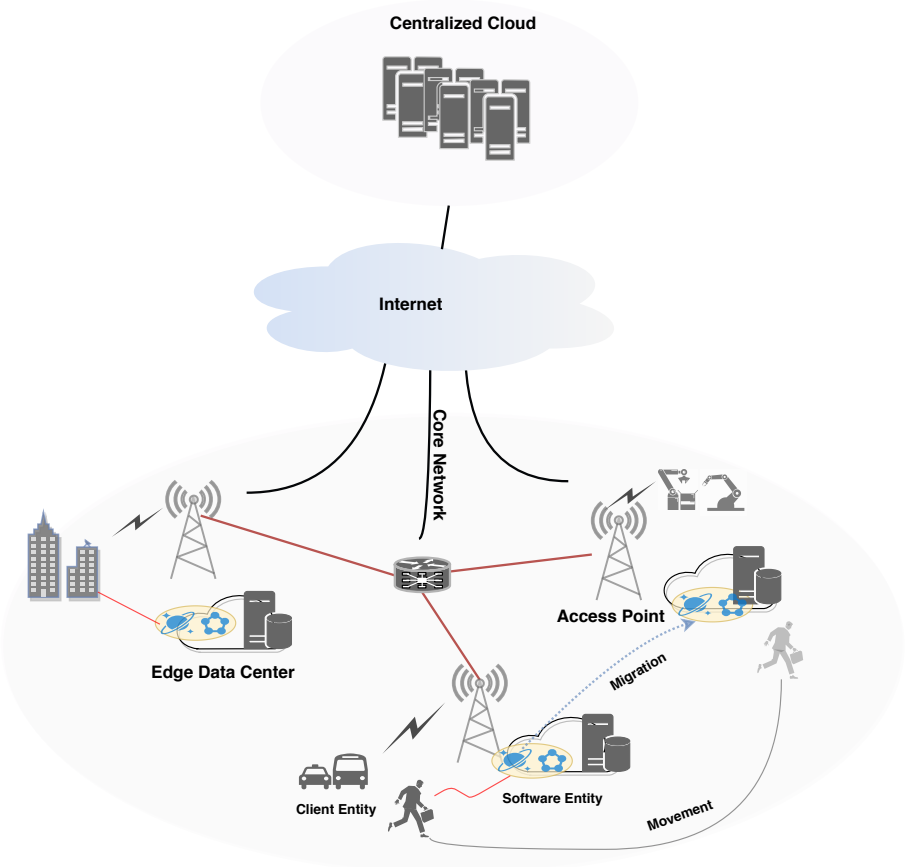


Figure 1.2: An illustrative MEC platform showing the association between client and service entities.

Many applications and services may need to exploit the resources of both MECs and distant clouds.

1.3 Research Problems and Objectives

MECs are emerging as novel computing platforms designed to overcome barriers to the success of new application types that we refer to as "MEC-native" applications. However, it is unrealistic to expect MECs to become successful based on these applications alone because MEC-native applications are unlikely to be extensively developed before MECs become widely available. MEC providers should therefore focus on the benefits MECs can offer to non-

MEC-native applications. A promising class of applications that may benefit greatly from deployment on MECs are cloud-native applications, particularly microservice-based applications with high deployment flexibility. Therefore, the first research objective of this thesis, **RO1**, was to answer the following two research questions:

1. *How much can cloud-native applications benefit from latency reduction when deployed on MECs?*

2. *How should cloud-native applications be engineered to maximize these benefits?*

Second, because they are designed to provide low-latency services and reduce network traffic to the central cloud, MECs basically attempt to provision end-users with resources from physically nearby EDCs. Therefore, the resource demand at each EDC depends heavily on the mobility behavior of nearby users. The number of end-users concurrently requiring services from a specific EDC may vary considerably. This user mobility together with the resource heterogeneity and wide geographical distribution of the infrastructure create new kinds of challenges in resource management. An important problem in the management of MECs is how to decide *where* the computation for each user should be performed, *what* resources should be allocated, and *how much* of each resource is needed, taking into account the unpredictability of user mobility behavior and the dynamic properties of the network. When a user requests a cloud service, that service may run either in the centralized cloud or in an MEC. Additionally, there may be multiple servers or datacenters within the centralized cloud or within individual MECs. It is therefore necessary to identify the optimal combination of resources to run the service. In addition, the user may move between geographical areas, so it is also important to decide whether and where to migrate the service as the user's location and/or the network state changes. The time taken to select and enact these resource management actions is important, especially when resource usage is likely to vary rapidly, as is common in MECs. Given these challenges, MECs require autonomous resource management systems that can continuously monitor workload dynamics and adapt to changes by continuously optimizing resource allocation. The acquired resources must be transparently provisioned and ready to use so as to meet users' expectations. Because of these needs, the second research objective, **RO2**, was to develop *an efficient workload prediction mechanism* that can understand the characteristics of a workload, anticipate the likely variation in workload at EDCs, and help the resource management operator to proactively identify and make important management decisions.

The main research objectives of this thesis are thus:

RO1 To quantify the benefits MECs can provide to non-MEC-applications.

RO2 To develop a workload prediction algorithm for Edge Data Centers.

1.4 Research Methodology

The work presented in this thesis primarily involved experiments on emulated and simulated systems.

To address **RO1**, we configured emulated Edge Clouds and studied the network delay between end-users, edge locations, and a distant centralized datacenter. To this end, we deployed two benchmark applications located on different resources, together with a workload generator to generate load on the servers. We then measured the response times of the deployed applications on the loaded servers.

To address **RO2**, we used machine learning techniques and statistical methods to develop a model for workload prediction in MECs. To evaluate the model, we emulated two MECs, one with a hexagonal topology and another based on the real geographical distribution of cellular base stations from a particular area. To simulate the load on the EDCs, we use two real mobility traces.

1.5 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 provides a brief overview of the main challenges facing MECs, introduces MAPE-K feedback loop autonomic resource management, and briefly reviews state-of-the-art techniques for resource management in MECs. Chapter 3 summarizes the contributions of each paper included in the thesis. Finally, Chapter 4 presents some suggestions for future work building on the results presented here.

Chapter 2

Resource Management in Mobile Edge Clouds

MECs are still in their infancy, and their infrastructure configuration has yet to be standardized. Therefore, most research on MECs has primarily focused on the concept of the MEC and its characteristics, as well as application scenarios [19, 20, 21]. In this chapter, we first describe the resource management challenges associated with MECs. We then introduce the MAPE-loop for autonomous resource management, which can be used to overcome these challenges. Finally, we present a literature review of previous efforts to address these challenges and develop different resource management mechanisms for MECs.

2.1 Resource Management Challenges

The key disruptive transformation of the MEC concept is the decentralization of the compute, storage, and networking resources of a cloud system and their redistribution towards the edge of the network, closer to the end-users, in "micro" data centers (i.e., data centers with lesser resource capabilities than a centralized cloud data center). This generates many benefits by reducing latency and mitigating bandwidth limits, unlocking the potential of new application types including IoT applications, autonomous vehicle systems, and AR/VR applications [14, 13]. However, the combination of the intrinsic characteristics of MECs with the inherent characteristics of clouds creates several challenges for resource management operators:

Highly Distributed and Heterogeneous Resource Capacity. The benefits MECs gain by moving computing resources toward the edge of the network are clear. However, the highly distributed and heterogeneous nature of MECs introduces difficult challenges in resource management. The new platform infrastructure may feature tens of large data centers and thousands of

micro data centers of various sizes colocated with radio base stations separated by 1 to 10 km. As a result, centralized strategies for monitoring system behavior and workload dynamics, and for resource allocation, may perform poorly in MECs despite being very efficient in centralized clouds.

User Mobility. To deliver low latency services and direct network traffic away from the central cloud, MECs seek to provision end-users with resources from EDCs located in the end-user's vicinity. The resource demand at each EDC therefore depends heavily on users' mobility behavior. The number of end-users concurrently requiring services from a specific EDC may exhibit large temporal fluctuations, causing load variation. The users' mobility behavior together with the inherent resource heterogeneity of MECs and the wide geographical distribution of the infrastructure create new challenges for resource management operators. The fundamentally intertwined questions of how many resources to allocate, where to place different application services among the available EDCs, and when to activate various resource management actions are inherently difficult to solve due to the scale, complexity, and dynamics of both infrastructure and applications.

More Flexibility in Deploying Software. Cloud applications are increasingly engineered as sets of multiple loosely-coupled fine-grained software components, each requiring different resources. To maximize the benefits of MECs, these components can be deployed on diverse resources ranging from centralized datacenters to edge locations. However, such deployment flexibility introduces significant challenges in analyzing, predicting and controlling resource allocations to optimize cost and energy efficiency while delivering the expected end-user Quality of Service.

2.2 A MAPE-K-based Autonomous Resource Management

The aim of resource management operators is to ensure that the reliability, availability, and performance targets of the MEC platform are met while minimizing costs and energy consumption. The challenges mentioned above make traditional centralized resource management strategies that rely on human intervention impractical [22, 23]. It is therefore be important to develop autonomic resource management strategies in which both the system's behavior and its workload dynamics are continuously monitored, and the monitoring data are used to automatically adjust resource allocations (in terms of both size and type) and the system's behavior in response.

An autonomic system is defined as a hierarchy of self-governing components, each of which consists of multiple interacting, autonomous components [24]. While the fundamental principles of autonomic systems are relatively well understood, the extreme scale, complexity, and dynamicism of MECs makes the practical implementation of those principles difficult [25].

The following sections introduce the individual components of a MAPE-K (Monitor, Analyze, Plan, Execution, and Knowledge) autonomic controller. To show how MAPE-K could be used to implement a dynamic adaptation policy and strategy, we present a proactive auto-scaler for MECs as a running example.

A proactive auto-scaler: Resource usage in datacenters fluctuates over time depending on end-users' demands. The purpose of an auto scaler is to dynamically adapt the resources assigned to an application depending on its workload so as to keep the response time for end user requests below some predefined target value while maximizing resource utilization. A major issue in resource provisioning is delay. For example, the average startup time of a Virtual Machine (VM) was determined to be 10 minutes [26], and containers hosting stateful applications may need several minutes [27] to replicate state until additional capacity is available to serve users. Because of this unavoidable startup time, a simple threshold-based reactive auto-scaler may be slow to converge to the capacity levels needed to meet current demand. It would therefore be desirable to have a proactive auto-scaler capable of predicting and provisioning the required resources in advance so they are ready to use within the user expectation.

To deliver low response time services, end-users of MECs are connected to EDCs located in their vicinity. The resource demand at individual EDCs will therefore fluctuate depending on user mobility. The proactive MAPE-K auto-scaler was designed to accommodate these fluctuations in MECs. This scaler is activated every 10 minutes to pro-actively provision/de-provision resources (e.g., CPUs) at each EDC. It also continuously monitors the system's workload and performance, using both as inputs to determine the number of CPUs that should be allocated to the hosted applications in order to maintain a low response time. The following subsections describe the individual components of a MAPE-K controller and their implementation in the proposed auto-scaler.

2.2.1 Monitor

The *Monitor* component samples the system's behavior and tracks the performance of running applications. The frequency of data acquisition by the *Monitor* component is predefined, typically on the basis of Shannon sampling theory [28]. Many monitoring systems and tools have been developed for dynamically monitoring systems with various objectives. For example, Amazon CloudWatch* is a monitoring and management service that gives developers, system operators, and IT managers actionable insights into running applications and system-wide performance. Similarly, NetLogger [29] is used to monitor and collect information on networks. GridEye [30] is a service-oriented monitoring system using in grid environments and other contexts. Prometheus [31] is an open source metrics-based monitoring system that is widely used in distributed systems. It provides a simple yet powerful data model based on time

*https://aws.amazon.com/cloudwatch/

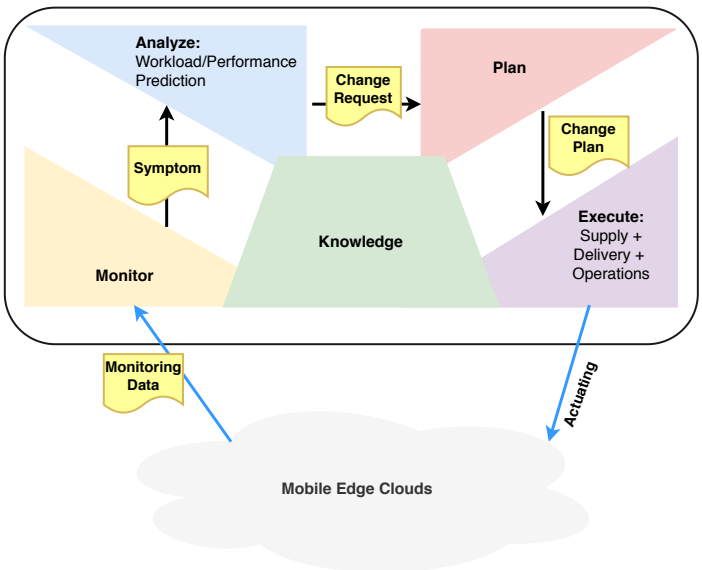


Figure 2.1: An MAPE-K loop Resource Management system in an MEC.

series and a flexible query language (PromQL) that facilitates analysis of the performance of deployed applications and computing infrastructure. Time series of different metrics are collected from endpoints that expose them and accessed using a pull model over HTTP. Monitoring the performance of complex distributed systems such as MECs and the applications running on them is difficult because of the unique characteristics of these systems. Brandon et al. presented FmonE [32], which is a lightweight, user-adjustable monitoring tool designed for MECs and Fog systems. FmonE uses a container orchestration system to create monitoring pipelines that are adapted to the unique features of an MEC's infrastructure.

In the auto-scaler, the *Monitor* component periodically gathers different metrics relating to the system and the current state of the hosted applications such as their workload, resource usage (e.g., CPU utilization per VM, memory usage, etc.) to facilitate analysis of the system and early detection of anomalies. The gathered data is stored in the form of time series, i.e., streams of timestamped values representing the same metric and the same set of labeled dimensions in the *Knowledge* database for further processing and analyzing by other components.

2.2.2 Analyze

The *Analyze* component provides the mechanisms used to observe and analyze situations to determine whether some change should be enacted [33]. To sup-

port decision-making by the *Plan* component, an autonomic manager must be able to perform complex data analysis and reason about the symptoms captured by the *Monitoring* component. To this end, the *Analyze* component uses complex models such as time series models, statistical models, and machine learning techniques to capture the static and dynamic characteristics of the system's hardware and software components, as well as the behavior of the real workload it processes.

In the case of the auto-scaler, the historical workload time series of EDCs recorded by the *Monitor* component are analyzed by using various statistical models and data mining techniques to extract recurring patterns related to workload characteristics, such as times when demand rises or falls during the day or over the course of the week. Based on these findings, an efficient workload prediction model is developed that reflects the dynamics of the workload, making it possible to anticipate variations in the workload of individual EDCs. Since the auto-scaler is activated once every 10 minutes, the predictive model must respond rapidly (e.g., within several seconds) and be very accurate to enable the *Plan* component to proactively plan for resource provisioning/deprovisioning in the EDCs. The predicted workload and the current state of the EDCs (e.g., their queue lengths, average response times, etc.) are the inputs that the performance modeler uses to estimate the resources (e.g., CPUs) required at each EDC in the next 10-minute window.

2.2.3 Plan

The *Plan* component is responsible for planning mitigation actions that will allow the managed system to adapt to predicted changes. Using the results generated by the *Analyze* component together with predefined target performance indicators relating to variables such as throughput and response times, the autonomic manager structures actions (e.g., admission control, resource allocation, migration, etc.) to ensure the system meets its performance targets while minimizing costs and energy consumption.

In our running example, the *Plan* component takes as input the estimated resource requirements of each EDC and converts this information into planned actions such as '*add x resource to EDC#1 and remove y resource from EDC#2*'.

2.2.4 Execute

The *Execute* component is responsible for scheduling and performing the planned adaptation actions. The execution of the plans also involves updating the *Knowledge* database that can be used by all components of the autonomic manager.

In our example, the *Execute* component uses some predefined APIs or middleware to communicate with the hardware or virtualization layer to enact the actions chosen by the *Plan* component.

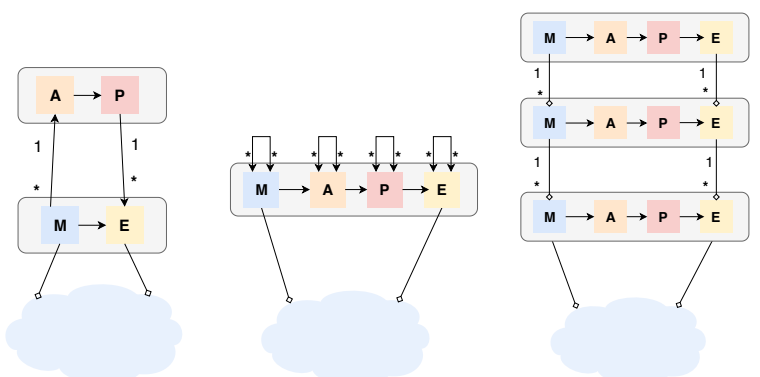
2.2.5 Knowledge

The *Knowledge* component stores data with an architected syntax and semantics, such as topological information, historical logs, policies, change requests, and change plans. In a complete loop, knowledge from other components is also stored. For example, the *Monitor* component generates knowledge about recent activities by logging the notifications it receives from a managed resource. Similarly, the *Execute* component might update the knowledge base with records of actions taken in response to the output of the *Analysis* and *Plan* components, making it possible to trace the actions' effects on the system. The *Knowledge* can be shared with all of the autonomic manager components mentioned above.

In the auto-scaler, the *Knowledge* component stores historical time series of the different metrics traced by the *Monitor* component, and also records the success or failure of previous decisions so as to improve the quality of subsequent decisions.

2.2.6 Multiple MAPE-K loops

In large, complex, and highly geo-distributed systems such as MECs, a fully centralized MAPE loop may be a poor solution because it could introduce a single point of failure and a bottleneck for scalability [25]. It is therefore necessary to investigate different MAPE patterns to design multiple MAPE loops that decentralize the components of each individual MAPE loop. Potentially suitable patterns include the *Master-slave* pattern, *Coordinated control* pattern, and the *Hierarchical control* pattern [34], as shown in Figure 2.2.



(a) Master–Slave pattern

(b) Coordinated Control pattern

(c) Hierarchical Control pattern

Figure 2.2: Different MAPE loop control patterns.

2.3 Review of the Literature on Resource Management in MECs

The characteristics of MECs present major challenges for resource management operators, as discussed above. This section reviews the efforts that have been made to address these challenges and to develop automated resource management mechanisms for MECs. We initially consider research on the *Capacity Sizing* problem, which requires the resource management operator to decide what type and quantity of resources should be reserved to meet an application's Quality of Service requirements. Second, we review work on the *Application and Workload Placement* problem, which is the problem of deciding where and when to deploy a service within the heterogeneous resource capacity of an MEC so as to achieve efficient resource utilization.

2.3.1 Capacity Sizing

One critical challenge facing the operators of any computing infrastructure is to consistently meet end-users' expectations while minimizing operational costs. The intertwined question of what and how many resources to allocate to each hosted application is not trivially answered. This is especially true for MECs, whose infrastructure makes these challenges much more severe than they are in conventional cloud systems. To solve this problem, it is necessary to think outside the box and employ concepts from multiple disciplines including feedback control loops, data analytics, and optimization techniques. A recent literature review [21, 35] highlighted the vast efforts that have been made in both academia and industry to solve the resource allocation problem.

Yin et al. proposed a task scheduling and resource allocation tool for delay-sensitive and high-concurrency applications in fog computing systems that is based on container technology [36]. This tool uses the delay constraints of the managed tasks to schedule and allocate resources from edge nodes or a centralized datacenter based on the objective of ensuring that the response times of the managed tasks remain below predefined thresholds. Chen et al. [37] proposed a framework consisting of a computation offloading mechanism and a joint communication and computation resource allocation method for the network operator. Based on predefined user ranking criteria, this framework can deliver performance guarantees for the managed applications.

Another effort to address the capacity sizing problem was presented by Mehta et al. [38], who developed a two-tier scheduler for allocating runtime resources to Industrial Internet of Things applications in MECs. A high-level scheduler is responsible for application admission and migration to meet long-term performance goals, while a low-level scheduler decides which application will occupy the runtime resources in the next execution period.

Using the concept of the MAPE feedback loop, Cardellini et al. [25] proposed a hierarchical decentralized resource allocation framework for data stream processing applications. The framework is based on a two-layered approach in

which timescale-related issues are handled separately from other concerns. The lower layer is responsible for controlling the adaptation of data stream processing operators by means of scaling and migration actions, while the higher layer is a centralized component that oversees general application performance.

Differently from these efforts, the approaches we present in Papers II and III begin by considering the correlation of changes in the workloads of neighboring EDCs in order to predict each EDC's workload in the near future. This approach yielded highly accurate predictions, showing that the proposed methods could be used to develop an efficient proactive auto-scaler to provision and de-provision resources in MECs as required to meet end-users' demands.

2.3.2 Application and Workload Placement

Cloud applications are increasingly engineered as sets of interacting components, each of which may require different kinds and quantities of resources to perform well. The increased deployment flexibility offered by MECs could in principle be very beneficial for such applications because their individual components could be deployed at different resource levels (ranging from the centralized datacenter to edge datacenters) provided that the application's overall performance goals are met. For example, a typical face recognition application will have face detection, image processing, feature extraction, and face recognition components. The face detection component is deployed on the end-user's device, the image processing and feature extraction components could be deployed at the edge datacenter, while the face recognition component could be deployed on the centralized distant datacenter. This distribution of components over available resources is a solution to the service placement problem for this hypothetical application. In general, the service placement problem is the problem of deciding where an application's services should be placed (and executed) within the hierarchy of the datacenter or cloud system; in the case of an MEC, each component of a cloud application could be placed anywhere from a centralized distant datacenter to an EDC near the user. The service placement problem in MECs is complicated by several factors not found in conventional clouds, including the limited coverage area of base stations, the dynamic nature of mobile users, and network background traffic. Nevertheless, it must be solved well because poor solutions can adversely affect the Quality of Service experienced by end users, potentially causing significant costs for both the application provider (due to unnecessary use of expensive resources) and the resource provider (as a consequence of repeatedly performing replacement actions due to poor initial placement decisions).

Tong et al. [39] attempted to solve the mobile workload placement problem in the hierarchical architecture of an edge cloud. They first designed a hierarchical edge cloud architecture that enables the aggregation of peak loads across various tiers of the edge cloud servers. An analytical model was then created to compare the efficiency of resource utilization between such hierarchical designs and a flat infrastructure. Additionally, to minimize the average program

execution delay, the authors developed an optimization algorithm that adaptively decides which edge cloud server a program should be deployed on and how much compute capacity should be allocated to it. Tarneberg et al. [40] presented a holistic algorithm for dynamically placing applications in MEC infrastructures. To minimize global system costs, the algorithm takes account of factors including the network link capacity, user expectations in term of latency, user mobility, and server provisioning costs. Taking a social Virtual Reality application as a potential "killer app" for emerging MECs, Wang et al. [41] introduced ITerative Expansion Moves (ITEM) to solve the combinatorial optimization problem for service entity placement. In [42], Wang et al. modeled users, a multi-component application, and physical MEC resources as graphs and considered service placement for a linear application graph with the goal of minimizing peak resource utilization for both compute resources and network links. To this end, the authors proposed online approximation algorithms for lacing tree application graphs onto tree physical graphs. Taking into account stochastic user mobility, Ouyang et al. [43, 44] proposed efficient heuristic algorithms to optimize long-term time-averaged migration costs. In [43], the authors proposed a novel mobility-aware online service placement framework to achieve a desirable balance between user latency and migration cost. Additionally, in [44], the authors proposed a joint service placement and routing algorithm designed to minimize total service placement costs.

All of the works discussed above focused primarily on MEC-native applications, i.e., applications engineered specifically to run on MECs. However, to encourage investment in MECs, it will be necessary to show that they can also benefit more diverse applications and services, such as cloud-native applications. To this end, the work presented in Paper I evaluates the performance of selected cloud-native applications when deployed on MECs using resources from various levels of the hierarchical infrastructure, ranging from the centralized datacenter to edge locations.

Chapter 3

Summary of Contributions

This thesis focuses on two main issues. The first is the potential for improving the performance of cloud applications by deploying them on MECs. MECs have emerged as distributed platforms that can complement existing cloud systems to overcome barriers to the success of MEC-native applications (e.g., IoT applications, autonomous vehicles, etc.). Much of the literature in this area focuses only on "killer apps" that could drive investment in MECs, such as IoT applications and augmented reality systems. However, given that the adoption of traditional clouds was fostered by legacy, non-cloud-native applications, we argue that MECs need also to provide benefits to non-MEC-native applications. Failing to do so risks creating a deadlock whereby infrastructure investment is slow due to a lack of MEC-native applications, and development of MEC-native applications is postponed until more MECs become available. Paper I addresses this issue by testing the potential for cloud applications to leverage the strengths of MECs to improve their performance in terms of end-to-end response time.

The second issue addressed in this thesis is the lack of reliable tools for workload prediction in MECs. To achieve the objectives of providing services with low latency and minimizing network traffic to the central cloud, MECs rely on the resources of EDCs located in close proximity to the end user. Therefore, the resource demand at any given EDC depends heavily on users' mobility behavior. The fundamentally intertwined questions of how much resource to allocate, where to place different application services among the available EDCs, and when to activate various resource management actions are inherently difficult to solve due to the scale, complexity, and dynamics of both MEC infrastructure and applications. To address this, we investigated the design and implementation of algorithms for efficient autonomous resource management in MECs. Papers II and III introduce two workload prediction algorithms that can be used to estimate the resource usage in EDCs in advance, enabling informed decision-making and the selection of effective management actions to ensure the EDCs consistently satisfy their Quality of Service (QoS) require-

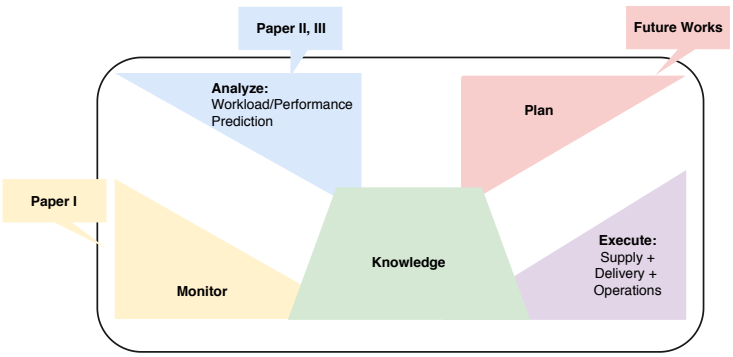


Figure 3.1: Contributions made by this thesis and future work to build on the results obtained.

ments while maximizing resource utilization.

3.1 Paper I

Chanh Nguyen, Amardeep Mehta, Cristian Klein, and Erik Elmroth. **Why Cloud Applications Are not Ready for the Edge (yet)**. *In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (SEC 2019), to appear, 2019.*

In Paper I, we address **RO1** by quantifying the benefits of deploying cloud-native applications on MECs. Two commonly cited potential benefits of MECs are lower latencies and lower core network bandwidth consumption. In this work we focus on latency because many end-user-facing cloud-native applications need low end-to-end response times; several studies have identified negative correlations between response times and revenues. To determine the impact of MEC deployment on latency, we emulated an MEC infrastructure with a distant datacenter and an edge datacenter. We focused on microservice-based applications because of their flexibility in deployment. Using two popular cloud benchmarks, SockShop and Web Serving, we empirically measured performance – specifically, end-to-end latency – under different deployment configurations, using resources from both distant datacenters and edge locations. Extensive experimentation revealed that against conventional wisdom, end-to-end latency does not improve significantly even when most services are deployed in an edge location. To explain these findings, we developed a network communication profiling tool and applied it to the two benchmarks to determine why they do not benefit from MEC development. It was found that these cloud-native applications tend to make many transactions between the user services and the corresponding database services when responding to end-user's requests. Consequently, deploying these services separately in different MEC layers causes poor application performance. This is an intrinsic problem that restricts the

scope for migrating such cloud-native applications to highly distributed environments such as MECs. We also investigated the communication patterns of current cloud-native application architectures to identify potential design improvements that would make it possible to take advantage of MECs. We addressed this problem at two levels: the application level and the network communication protocol level.

I was the main author; I contributed to the formulation of the problem, conducted the experiments, and helped write the paper. Amardeep Mehta helped design the Web Serving experiments and wrote the section of the paper dealing with the Web Serving results. Cristian Klein and Erik Elmroth had advisory roles that included discussions about the problem formulation, methods, experiments, and the presentation of the results.

3.2 Paper II

Chanh Nguyen, Cristian Klein, and Erik Elmroth. **Location-aware load prediction in Edge Data Centers**. *In Proceedings of the 2nd IEEE International Conference on on Fog and Mobile Edge Computing (FMEC), pp. 25-31, IEEE, 2017.*

In MECs, the operator's ability to perform capacity adjustment and planning is complicated by the bounded coverage radius of the base station, the limited capacity of each EDC, and the mobility of users. It would therefore be highly desirable to develop a self-managed system for MECs efficiently decides how much scaling is needed, when it should be activated, and where to place and migrate services. However, such a system would require an accurate and reliable method of predicting the characteristics of the MEC's workload, including its variation in time and space.

In Paper II, we address **RO2** by proposing a location-aware workload prediction tool. The fact that EDCs are located in the near vicinity of users means that changes in the workloads of nearby EDCs may be strongly correlated (for example, when a user moves from the area served by one EDC to an area served by another, the first EDC's workload will fall while that of the other will increase). This information could in principle be exploited to improve the accuracy of load prediction in MECs. The developed tool therefore predicts the load of each individual EDC based on its own historical load time-series (as is done for centralized clouds) as well as those of its neighboring EDCs. This is done using the Vector Auto Regression (VAR) Model, which exploits the correlations between the load time-series of adjacent EDCs.

To evaluate our approach, we used real world mobility traces for taxis in San Francisco, USA to simulate the load in each EDC. We emulated a MEC platform consisting of a cellular infrastructure of 37 cells arranged in a hexagonal grid covering the area of San Francisco. Each cell contained one EDC providing services to all end-users within that cell. Our proposed algorithm achieved an average accuracy of 93% in the experiments, outperforming the state-of-the-art

alternative by 4.3%. Given the scale of MECs, such an improvement in predictive performance could yield significant gains in the efficiency of resource allocation, and thus substantial cost savings.

I was the main author; I contributed to the formulation of the problem, conducted the experiments, and wrote the paper. Cristian Klein and Erik Elmroth had advisory roles that included discussions regarding problem formulation, methods, experiments, and presentation of results.

3.3 Paper III

Chanh Nguyen, Cristian Klein, and Erik Elmroth. **Multivariate Long Short-term Memory based Location-aware load prediction in Edge Data Centers**. *In Proceedings of the 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (IEEE/ACM CCGrid 2019), pp. 341-350, IEEE/ACM, 2019.*

Paper III also addresses **RO2** by building on the tool proposed in Paper II, which uses the correlation between the workload fluctuations of neighboring EDCs to improve predictive accuracy. An alternative location-aware workload prediction tool for EDCs that uses Long Short-Term Memory (LSTM) networks is presented. In essence, LSTM networks are special recurrent neural networks that incorporate integrated multiplicative nonlinear gate units with a linear dependence between memory cell states. They can capture the temporal dependencies of time series and have a high rate of learning per time step, making them well suited for predicting the workload of EDCs. To predict the workload of individual EDCs, we built an LSTM-based network that takes as input the multivariate workload time series of the EDCs in the vicinity of the predicted EDC.

Although the background and problem definition of this paper are identical to those for Paper II, the new method offers superior predictive accuracy to that reported in the earlier paper. Additionally, the new method differs from the earlier one in three important ways: 1) it relies a neural network-based technique, 2) it was tested in an extensive series of experiments using two real mobility traces to simulate the workload of EDCs, together with data on the real geographical locations of network base stations (emulating an MEC infrastructure in which the locations of the EDCs match those of the real network base stations); and 3) its predictive performance was validated using an input-shaking approach.

In evaluations based on the first of the real mobility traces mentioned above, the normalized root mean square error (NRMSE) observed with the neural network-based method proposed in Paper III was 17% lower than that for the location-aware method presented in Paper II and 44% lower than that for a location-unaware method previously reported in the literature; the corresponding values in evaluations using the second real mobility trace were 12% and 41%, respectively. Additionally, sensitivity analyses using different input shak-

ing techniques clearly demonstrated that the neural network-based method is stable and robust.

I was the main author; I contributed to the formulation of the problem, conducted the experiments, and wrote the paper. Cristian Klein and Erik Elmroth had advisory roles that included discussions regarding problem formulation, methods, experiments, and presentation of results.

Chapter 4

Future Work

The studies described in the preceding section sought to address different fundamental resource management challenges associated with MECs. First, to spur the development of MECs, we quantified their potential to enhance the performance of cloud-native applications. These studies showed that additional engineering work is needed to adapt existing cloud-native applications to MEC-like environments. We also introduced two workload prediction models for MECs that exploit the correlation between workload changes in neighboring EDCs. The high predictive accuracy achieved with these models suggests that they could be used effectively in a full resource management operation loop to improve capacity sizing planing in MECs. As discussed in Chapter 2, one of the major challenges facing potential MEC operators is their heterogeneous resource distribution, which makes centralized resource management strategies impractical because they introduce single points of failure. Decentralized autonomic strategies are thus preferable. Consequently, future efforts in this area will focus on developing autonomic resource management systems for MECs. This will be done by extending the work presented here in three distinct directions to address three main problems:

First, *elasticity* is an important feature of cloud computing that MECs must share to attract end-users. It is therefore essential to find ways of letting MECs dynamically adjust their resource allocations to meet changing workload demands. The workload prediction methods proposed in Papers II and III achieve high predictive accuracies and could therefore potentially be used to develop *an elastic control framework for MECs*. The aim is to create a management system that is adaptive and self-optimizing with respect to workload changes.

Second, to create a full MAPE-loop based autonomic resource management system, we want to focus on *planning* with the aim of developing an efficient solution for identifying (near-) optimal management actions including *resource scheduling and planning*. Specifically, we intend to develop a decentralized self-managed system based on models of application performance and workload

characteristics as well as real-time analytics of run-time monitoring data. This system will optimize capacity allocations in terms of size, type, and location, using feedback control and optimization to ensure that reliability, availability, and performance targets are met while minimizing costs.

Third, modern applications are increasingly architected as collections of many components, each with different resource requirements (e.g., some components are highly compute intensive, while others require more bandwidth). Such multi-component applications present non-trivial service placement problems when combined with the heterogeneous resource capacity of MECs; it is essential to deploy each component in a way that ensures satisfactory performance of the application as a whole. Furthermore, the mobility of users in some settings (e.g., autonomous vehicles, users of augmented reality assistance applications, streaming video, etc.) makes resource management much more challenging because it introduces the possibility of large and rapid changes in resource demand in individual EDCs. Therefore, to develop *an optimal service placement solution*, it is essential to consider the *stochasticity of user mobility* to decide where to place the services so as to minimize both placement and migration costs.

Bibliography

[1] SM Riazul Islam, Daehan Kwak, MD Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. "The Internet of Things for Health Care: A Comprehensive Survey". In: *IEEE Access* 3 (2015), pp. 678–708. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2015.2437951.

[2] Sabina Jeschke, Christian Brecher, Tobias Meisen, Denis Özdemir, and Tim Eschert. "Industrial Internet of Things and Cyber Manufacturing Systems". In: Cham: Springer International Publishing, 2017, pp. 3–19. ISBN: 978-3-319-42559-7. DOI: 10.1007/978-3-319-42559-7_1.

[3] Saeed Asadi Bagloee, Madjid Tavana, Mohsen Asadi, and Tracey Oliver. "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies". In: *Journal of Modern Transportation* 24.4 (Dec. 2016), pp. 284–303. ISSN: 2196-0577. DOI: 10.1007/s40534-016-0117-3.

[4] Edgar Alan Rayo. *Artificial Intelligence at Disney, Viacom, and Other Entertainment Giants*. https://www.techemergence.com/ai-at-disney-viacom-and-other-entertainment-giants/. Feb. 2018. (Visited on 01/01/2019).

[5] The 5G Infrastructure Association. *5G Vision: The 5G Infrastructure Public Private Partnership:the next generation of communicationnetworks and services*. https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf. Feb. 2015. (Visited on 01/01/2019).

[6] Ericsson Corporate Communications. *Internet of Things to overtake mobile phones by 2018: Ericsson Mobility Report*. https://www.ericsson.com/en/press-releases/2016/6/internet-of-things-to-overtake-mobile-phones-by-2018-ericsson-mobility-report. June 2016. (Visited on 01/01/2019).

[7] *Cisco Global Cloud Index: Forecast and Methodology,2016–2021*. https://www.techemergence.com/ai-at-disney-viacom-and-other-entertainment-giants/. 2018. (Visited on 01/01/2019).

[8] Peter C. Lincoln. "Low Latency Displays for Augmented Reality". PhD thesis. University of North Carolina at Chapel Hill, 2017.

[9] Per Skarin, William Tärneberg, Karl-Erik Årzén, and Maria Kihl. "Towards Mission-Critical Control at the Edge and Over 5G". In: *CoRR* abs/1803.02123 (2018). arXiv: 1803.02123.

[10] Karthik Kumar and Yung-Hsiang Lu. "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" In: *Computer* 43.4 (Apr. 2010), pp. 51–56. ISSN: 0018-9162. DOI: 10.1109/MC.2010.98.

[11] Peter Mell and Tim Grance. "The NIST definition of cloud computing". In: *Communications of the ACM* 53 (Jan. 2011). DOI: 10.6028/NIST.SP.800-145.

[12] Mahadev Satyanarayanan. "A Brief History of Cloud Offload: A Personal Journey from Odyssey Through Cyber Foraging to Cloudlets". In: *GetMobile: Mobile Comp. and Comm.* 18.4 (Jan. 2015), pp. 19–23. ISSN: 2375-0529. DOI: 10.1145/2721914.2721921.

[13] Zhuo Chen, Wenlu Hu, Junjue Wang, Siyan Zhao, Brandon Amos, Guanhang Wu, Kiryong Ha, Khalid Elgazzar, Padmanabhan Pillai, Roberta Klatzky, Daniel Siewiorek, and Mahadev Satyanarayanan. "An Empirical Study of Latency in an Emerging Class of Edge Computing Applications for Wearable Cognitive Assistance". In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. SEC '17. San Jose, California: ACM, 2017, 14:1–14:14. ISBN: 978-1-4503-5087-7. DOI: 10.1145/3132211.3134458.

[14] Wenlu Hu, Ying Gao, Kiryong Ha, Junjue Wang, Brandon Amos, Zhuo Chen, Padmanabhan Pillai, and Mahadev Satyanarayanan. "Quantifying the Impact of Edge Computing on Mobile Applications". In: *Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems*. APSys '16. Hong Kong, Hong Kong: ACM, 2016, 5:1–5:8. ISBN: 978-1-4503-4265-0. DOI: 10.1145/2967360.2967369.

[15] Mahadev Satyanarayanan, Grace Lewis, Edwin Morris, Soumya Simanta, Jeff Boleng, and Kiryong Ha. "The Role of Cloudlets in Hostile Environments". In: *IEEE Pervasive Computing* 12.4 (Oct. 2013), pp. 40–49. ISSN: 1536-1268. DOI: 10.1109/MPRV.2013.77.

[16] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. "Fog Computing and Its Role in the Internet of Things". In: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. MCC '12. Helsinki, Finland: ACM, 2012, pp. 13–16. ISBN: 978-1-4503-1519-7. DOI: 10.1145/2342509.2342513.

[17] Tarik Taleb and Adlen Ksentini. "Follow me cloud: interworking federated clouds and distributed mobile networks". In: *IEEE Network* 27.5 (Sept. 2013), pp. 12–19. ISSN: 0890-8044. DOI: 10.1109/MNET.2013.6616110.

[18] João Soares, Carlos Gonçalves, Bruno Parreira, Paulo Tavares, Jorge Carapinha, João Paulo Barraca, Rui L Aguiar, and Susana Sargento. "Toward a telco cloud environment for service functions". In: *IEEE Communications Magazine* 53.2 (Feb. 2015), pp. 98–106. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7045397.

[19] Arif Ahmed and Ejaz Ahmed. "A survey on mobile edge computing". In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. Jan. 2016, pp. 1–8. DOI: 10.1109/ISCO.2016.7727082.

[20] Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. "Survey on Multi-Access Edge Computing for Internet of Things Realization". In: *IEEE Communications Surveys Tutorials* 20.4 (Fourthquarter 2018), pp. 2961–2991. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2849509.

[21] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. "All one needs to know about fog computing and related edge computing paradigms: A complete survey". In: *Journal of Systems Architecture* (2019). ISSN: 1383-7621. DOI: 10.1016/j.sysarc.2019.02.009.

[22] William Tärneberg, Amardeep Mehta, Johan Tordsson, Maria Kihl, and Erik Elmroth. "Resource Management Challenges for the Infinite Cloud". English. In: (2015). 10th International Workshop on Feedback Computing at CPSWeek 2015 ; Conference date: 13-04-2015.

[23] Constantin Adam and Rolf Stadler. "Service Middleware for Self-Managing Large-Scale Systems". In: *IEEE Transactions on Network and Service Management* 4.3 (Dec. 2007), pp. 50–64. ISSN: 1932-4537. DOI: 10.1109/TNSM.2007.021103.

[24] Jeffrey O Kephart and David M Chess. "The vision of autonomic computing". In: *Computer* 36.1 (Jan. 2003), pp. 41–50. ISSN: 0018-9162. DOI: 10.1109/MC.2003.1160055.

[25] Valeria Cardellini, Francesco Lo Presti, Matteo Nardelli, and Gabriele Russo Russo. "Decentralized self-adaptation for elastic Data Stream Processing". In: *Future Generation Computer Systems* 87 (2018), pp. 171–185. ISSN: 0167-739X. DOI: 10.1016/j.future.2018.05.025.

[26] Zach Hill, Jie Li, Ming Mao, Arkaitz Ruiz-Alvarez, and Marty Humphrey. "Early observations on the performance of Windows Azure". In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. ACM. 2010, pp. 367–376.

[27] Hiep Nguyen, Zhiming Shen, Xiaohui Gu, Sethuraman Subbiah, and John Wilkes. "{AGILE}: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service". In: *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13)*. 2013, pp. 69–82.

[28] Abdul J. Jerri. “The Shannon sampling theorem—Its various extensions and applications: A tutorial review”. In: *Proceedings of the IEEE* 65.11 (Nov. 1977), pp. 1565–1596. ISSN: 0018-9219. DOI: 10.1109/PROC.1977.10771.

[29] Dan Gunter, Brian Tierney, Brian Crowley, Mason Holding, and Jason Lee. “NetLogger: a toolkit for distributed system performance analysis”. In: *Proceedings 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (Cat. No.PR00728)*. Aug. 2000, pp. 267–273. DOI: 10.1109/MASCOT.2000.876548.

[30] Wei Fu and Qian Huang. “GridEye: A Service-oriented Grid Monitoring System with Improved Forecasting Algorithm”. In: *2006 Fifth International Conference on Grid and Cooperative Computing Workshops*. Oct. 2006, pp. 5–12. DOI: 10.1109/GCCW.2006.51.

[31] Brian Brazil. *Prometheus: Up & Running: Infrastructure and Application Performance Monitoring*. ” O’Reilly Media, Inc.”, 2018.

[32] Alvaro Brandon, Maria S Perez, Jesus Montes, and Alberto Sanchez. “Fmone: A flexible monitoring solution at the edge”. In: *Wireless Communications and Mobile Computing* 2018 (2018).

[33] IBM. *An architectural blueprint for autonomic computing*. 2006. URL: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf (visited on 01/01/2019).

[34] Danny Weyns, Bradley Schmerl, Vincenzo Grassi, Sam Malek, Raffaela Mirandola, Christian Prehofer, Jochen Wuttke, Jesper Andersson, Holger Giese, and Karl M Göschka. “On patterns for decentralized control in self-adaptive systems”. In: *Software Engineering for Self-Adaptive Systems II*. Springer, 2013, pp. 76–107. ISBN: 978-3-642-35813-5. DOI: 10.1007/978-3-642-35813-5_4.

[35] Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. “Survey on Multi-Access Edge Computing for Internet of Things Realization”. In: *IEEE Communications Surveys Tutorials* 20.4 (Fourthquarter 2018), pp. 2961–2991. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2849509.

[36] Luxiu Yin, Juan Luo, and Haibo Luo. “Tasks Scheduling and Resource Allocation in Fog Computing Based on Containers for Smart Manufacturing”. In: *IEEE Transactions on Industrial Informatics* 14.10 (Oct. 2018), pp. 4712–4721. ISSN: 1551-3203. DOI: 10.1109/TII.2018.2851241.

[37] Xu Chen, Wenzhong Li, Sanglu Lu, Zhi Zhou, and Xiaoming Fu. “Efficient Resource Allocation for On-Demand Mobile-Edge Cloud Computing”. In: *IEEE Transactions on Vehicular Technology* 67.9 (Sept. 2018), pp. 8769–8780. ISSN: 0018-9545. DOI: 10.1109/TVT.2018.2846232.

[38] Amardeep Mehta, Ewnetu Bayuh Lakew, Johan Tordsson, and Erik Elmroth. "Utility-based Allocation of Industrial IoT Applications in Mobile Edge Clouds". In: *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. Nov. 2018, pp. 1–10. DOI: 10.1109/PCCC.2018.8711075.

[39] Liang Tong, Yong Li, and Wei Gao. "A hierarchical edge cloud architecture for mobile computing". In: *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. Apr. 2016, pp. 1–9. DOI: 10.1109/INFOCOM.2016.7524340.

[40] William Tärneberg, Amardeep Mehta, Eddie Wadbro, Johan Tordsson, Johan Eker, Maria Kihl, and Erik Elmroth. "Dynamic application placement in the Mobile Cloud Network". In: *Future Generation Computer Systems* 70 (2017), pp. 163–177. ISSN: 0167-739X. DOI: doi.org/10.1016/j.future.2016.06.021.

[41] Lin Wang, Lei Jiao, Ting He, Jun Li, and Max Muhlhauser. "Service Entity Placement for Social Virtual Reality Applications in Edge Computing". In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. Apr. 2018, pp. 468–476. DOI: 10.1109/INFOCOM.2018.8486411.

[42] Shiqiang Wang, Murtaza Zafer, and Kin K. Leung. "Online Placement of Multi Component Applications in Edge Computing Environments". In: *IEEE Access* 5 (2017), pp. 2514–2533. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2665971.

[43] Tao Ouyang, Zhi Zhou, and Xu Chen. "Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing". In: *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. June 2018, pp. 1–10. DOI: 10.1109/IWQoS.2018.8624174.

[44] Amir Varasteh, Sandra Hofmann, Nemanja Deric, Mu He, Dominic Schupke, Wolfgang Kellerer, and Carmen Mas Machuca. "Mobility-Aware Joint Service Placement and Routing in Space-Air-Ground Integrated Networks". In: *CoRR* abs/1902.01682 (2019). arXiv: 1902.01682.