UMEÅ UNIVERSITY

Location-aware Resource Allocation in Mobile Edge Clouds

Chanh Nguyen

Doctoral Thesis, February 2021
Department of Computing Science
Umeå University
Sweden

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

`https://people.cs.umu.se/chanh/`
chanh@cs.umu.se

ISBN 978-91-7855-467-6
ISSN 0348-0542
UMINF 21.01

Printed by Cityprint i Norr AB, Umeå, 2021

Abstract

Over the last decade, cloud computing has realized the long-held dream of computing as a utility, in which computational and storage services are made available via the Internet to anyone at any time and from anywhere. This has transformed Information Technology (IT) and given rise to new ways of designing and purchasing hardware and software. However, the rapid development of the Internet of Things (IoTs) and mobile technology has brought a new wave of disruptive applications and services whose performance requirements are stretching the limits of current cloud computing systems and platforms. In particular, novel large scale mission-critical IoT systems and latency-intolerant applications strictly require very low latency and strong guarantees of privacy, and can generate massive amounts of data that are only of local interest. These requirements are not readily satisfied using modern application deployment strategies that rely on resources from distant large cloud datacenters because they easily cause network congestion and high latency in service delivery. This has provoked a paradigm shift leading to the emergence of new distributed computing infrastructures known as Mobile Edge Clouds (MECs) in which resource capabilities are widely distributed at the edge of the network, in close proximity to end-users. Experimental studies have validated and quantified many benefits of MECs, which include considerable improvements in response times and enormous reductions in ingress bandwidth demand. However, MECs must cope with several challenges not commonly encountered in traditional cloud systems, including user mobility, hardware heterogeneity, and considerable flexibility in terms of where computing capacity can be used. This makes it especially difficult to analyze, predict, and control resource usage and allocation so as to minimize cost and maximize performance while delivering the expected end-user Quality-of-Service (QoS). Realizing the potential of MECs will thus require the design and development of efficient resource allocation systems that take these factors into consideration.

Since the introduction of the MEC concept, the performance benefits achieved by running MEC-native applications (i.e., applications engineered specifically for MECs) on MECs have been clearly demonstrated. However, the benefits of MECs for non-MEC-native applications (i.e., application not specifically engineered for MECs) are still questioned. This is a fundamental issue that must be explored because it will affect the incentives for service providers and application developers to invest in MECs. To spur the development of MECs, the first part of this thesis presents an extensive *investigation of the benefits that MECs can offer to non-MEC-native applications.* One class of non-MEC-native applications that could potentially benefit significantly from deployment on an MEC is cloud-native applications, particularly micro-service-based applications with high deployment flexibility. We therefore quantitatively compared the performance of cloud-native applications deployed using resources from cloud datacenters and edge locations. We then developed a network communication profiling tool to identify aspects of these applications that reduce the benefits derived from deployment on MECs, and proposed design improvements that would allow such applications to better exploit MECs' capabilities.

The second part of this thesis addresses problems related to resource allocation in highly distributed MECs. First, to overcome challenges arising from the dynamic nature of resource demand in MECs, we used statistical time series models and machine learning techniques to develop two *location-aware workload prediction models* for EDCs that account for both user mobility and the correlation of workload changes among EDCs in close physical proximity. These models were then utilized to develop an *elasticity controller for MECs*. In essence, the controller helps MECs to perform resource allocation, i.e. to answer the intertwined questions of what and how many resources should be allocated and when and where they should be deployed.

The third part of the thesis focuses on problems relating to *the real-time placement of stateful applications on MECs*. Specifically, it examines the questions of where to place applications so as to minimize total operating costs while delivering the required end-user QoS and whether the requested applications should be migrated to follow the user's movements. Such questions are easy to pose but intrinsically hard to answer due to the scale and complexity of MEC infrastructures and the stochastic nature of user mobility. To this end, we first thoroughly modeled the workloads, stateful applications, and infrastructures to be expected in MECs. We then formulated the various costs associated with operating applications, namely the resource cost, migration cost, and service quality degradation cost. Based on our model, we proposed two online application placement algorithms that take these factors into account to minimize the total cost of operating the application.

The methods and algorithms proposed in this thesis were evaluated by implementing prototypes on simulated testbeds and conducting experiments using workloads based on real mobility traces. These evaluations showed that the proposed approaches outperformed alternative state-of-the-art approaches and could thus help improve the efficiency of resource allocation in MECs.

Sammanfattning

Under det senaste årtiondet har datormoln förverkligat den långvariga drömmen att tillhandahålla datorkapacitet som en tjänst, där beräknings- och lagringstjänster är tillgängliga via Internet till vem som helst, när som helst och från var som helst. Detta har förändrat informationsteknik (IT) och givit upphov till nya sätt att designa och köpa hårdvara och mjukvara. Den snabba utvecklingen av Internet of Things (IoTs) och mobila teknologier har lett till en ny våg av innovativa applikationer och tjänster vars prestandakrav tänjer på molntjänsters och plattformars nuvarande begränsningar.

Nya, storskaliga och uppdragskritiska IoT-system och latenskänsliga applikationer kräver låg latens och starka garantier för integritet och kan generera massiva mängder data som enbart är av lokalt intresse, nära där de genererades. Dessa krav är svåra att uppfylla då moderna moderna strategier för driftsättning av applikationer används, då dessa ofta baseras på resurser belägna i stora avlägsna datacenter som ofta orsakar övertrafikerade nätverk och hög latens vid leverans av tjänster. Detta har orsakat ett paradigmskifte som har lett till framväxten av ny infrastruktur för distribuerade system känd som Mobile Edge Clouds (MECs), där resurser kan distribueras till kanten av nätverket, i nära anslutning till slutanvändare. Experimentella studier har validerat och kvantifierat många fördelar med MECs, inklusive avsevärda förbättringar i responstider och enorma reduceringar i bandbreddskrav. MECs måste däremot hantera flera utmaningar som vanligtvis inte stöts på i vanliga molnsystem, inklusive användarmobilitet, hårdvaruheterogenitet och avsevärd flexibilitet var beräkningskapacitet kan användas. Detta gör det speciellt svårt att analysera, förutsäga och kontrollera resursanvändning och allokering för att minimera kostnader och maximera prestanda samtidigt som den förväntade Quality-of-Service (QoS) levereras. Att realisera MECs potential kräver därför design och utveckling av effektiva resursallokeringssytem som tar hänsyn till dessa faktorer.

Sedan introduktionen av MEC-konceptet har presetandafördelar med att köra MEC-native applikationer (d.v.s. applikationer konstruerade specifikt för MECs) på MECs tydligt påvisats. Fördelar av MECs för icke-MEC-native applikationer (d.v.s. applikationer som inte är speciellt konstruerade för MECs) går däremot fortfarande att ifrågasätta. Detta är ett fundamentalt problem som måste utforskas då det kommer påverka incitament från tjänsteleverantörer och applikationsutvecklare att investera i MECs. För att sporra utvecklingen av MECs presenterar första delen av denna avhandling en omfattande utredning av fördelarna som MECs kan erbjuda för icke-MEC-native applikationer. En klass av icke-MEC-native applikationer som potentiellt skulle kunna dra fördel av driftsättning på ett MEC är cloud-native applikationer, i synnerhet microservice-baserade applikationer med hög driftsättningsflexibilitet. Vi utförde därför en kvantitativ jämförelse av prestandan hos cloud-native applikationer som var driftsatta med resurser i molndatacenter och platser belägna på nätverkets utkant. Vi utvecklade därefter ett profileringsverktyg för nätverkskommunikation för att identifiera aspekter hos dessa applikationer som minskar fördelarna erhållna vid driftsättningen på MECs, och föreslog designförbättringar som tillåter sådana applikationer att bättre nyttja MECs potential och möjligheter.

Andra delen av denna avhandling adresserar problemet med resursallokering i högt distribuerade MECs. För att överkomma de utmaningar relaterade till den dynamiska karaktären hos MECs resursefterfrågan, använde vi statistiska tidsseriemodeller och maskininlärningstekniker för att utveckla två platsmedvetna modeller för att förutsäga arbetsbelastningen hos datorresurser vid kanten av nätverket som tar hänsyn till användarmobilitet och korrelationen mellan förändringar i arbetsbelastningen mellan nära belägna datorresurser. Dessa modeller används sedan för att utveckla en elasticitetsregulator för MECs. I huvudsak hjälper regulatorn MECs att utföra resursallokering, d.v.s. att besvara de sammanflätade frågorna om vilka och hur många resurser som ska allokeras och när och var de ska driftsättas.

Den tredje delen av denna avhandling fokuserar på problem som relaterar till placering, i realtid, av MEC-applikationer som kräver att information om användarens session sparas mellan dess interaktioner. Den undersöker särskilt frågorna var applikationer ska placeras för att minimera de totala driftkostnaderna samtidigt som efterfrågad QoS upprättshålls och huruvida the efterfrågade applikationerna ska migreras för att följa användarens förflyttningar. Sådana frågor är lätta att ställa men fundamentalt svåra att besvara på grund av MEC-infrastrukturens storskalighet och komplexitet och den stokastiska karaktären hos användarmobilitet. För detta ändamål utformade vi modeller för arbetsbelastningar, applikationer samt hårdvara som kan förväntas i MECs. Därefter formulerade vi olika kostnader associerade med applikationsdrift, nämligen resurskostnader, flyttkostnader och kostnader för försämrad av servicenivå. Baserat på vår modell, föreslår vi två online applikationsplaceringsalgoritmer som tar hänsyn till dessa faktorer för att minimera de totala driftkostnaderna för applikationen.

De föreslagna metoderna och applikationerna i denna avhandling har utvärderats genom att implementera prototyper på simulerade testbäddar och genomföra experiment med arbetsbelastning baserade på riktiga mobilitetsdata. Dessa utvärderingar visade att de föreslagna tillvägagångssätten överträffar alternativa moderna tillvägagångssätt och kan således hjälpa till att förbättra effektiviteten för resursallokering i MECs.

Preface

This thesis includes a brief introduction to Mobile Edge Clouds (MECs), a discussion on the challenges and problems of resource allocation in MECs, a summary of the contributions made in the five included papers, and some suggestions for future research directions. The contributions of this thesis are presented in detail in the following five included papers[†]:

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'19)*, ACM, pp. 250-263, 2019.

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

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 (CCGRID)*, IEEE, pp. 341-350, 2019.

Paper IV **Chanh Nguyen**, Cristian Klein, and Erik Elmroth. Elasticity Control for Latency-Intolerant Mobile Edge Applications. *In Proceedings of the 5th ACM/IEEE Symposium on Edge Computing (SEC'20)*, ACM, pp. 70-83, 2020.

Paper V **Chanh Nguyen**, Cristian Klein, and Erik Elmroth. State-aware Application Placement in Mobile Edge Clouds. *Submitted for journal publication*, 2021.

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

In addition to the papers included in this thesis, the following article has been produced during the doctoral studies:

- Thang Le Duc, **Chanh Nguyen**, and Per-Olov Östberg. Towards Proactive Resource Allocation for Large-Scale Applications in Cloud-Edge Computing Environments. *Submitted for journal publication*, 2021.

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

Acknowledgments

"It was my luck to have a few good teachers in my youth, men and women who came into my dark head and lit a match."

— Yann Martel, Life of Pi

My Ph.D. journey has been an enriching experience that has enlightened me in uncountable ways. On this journey, I have walked alongside several people who have had many positive impacts on my mindset and life goals. It is my duty to acknowledge every such person.

First of all, Prof. Erik Elmroth – my main supervisor. I offer my profoundest gratitude to you for giving me the tremendous opportunity to pursue my studies in your group. I would like to thank you for your continuous guidance, support, and encouragement throughout both the rough and the enjoyable moments of my Ph.D. work. I still remember the very first lesson you gave me on the first day of my Ph.D. studentship. By telling me a story about a Nobel laureate, you showed me that the most important characteristic required for a researcher is *patience*. And you then reinforced the lesson by showing me day by day what patience means by persistently supporting me throughout my research. Under your wings, I have not only learned to become a researcher, but also how to become a real man by following your example.

Secondly, Dr. Cristian Klein – my co-supervisor. You are to me a teacher, an endless and profound source of knowledge, an older brother who always offers support, an empathic and encouraging friend, and a fabulous search engine with answers to problems concerning both research and the outside world. Your wisdom in virtually every domain just lights me up whenever I talk to you. From you I learned how a research work should be developed, how to present a scientific work in a concise but easily understood way, and so on. You encouraged me to submit my research works; without your influence, they would still sit unpublished on my desk. Because of that and more, I would like to offer my sincere gratitude to you. Until the later parts of my Ph.D. studies, I did not reach out to you as much as I should have, which is something I deeply regret! Officially, I'm your first Ph.D. student; I hope I did not make you feel despondent and tired as an academic supervisor.

I would also like to express my appreciation to Prof. Frank Drewes and Prof. Bo Kågström for your constructive comments and suggestions on my Individual Study Plan throughout the study.

Additional thanks are owed to all the past and current members of ADSlab and my colleagues from the Department of Computing Science at Umeå University for your support and for the times we shared during lunches, fikas, and other enjoyable events.

This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), which is funded by the Knut and Alice Wallenberg Foundation. WASP provided both funding and access to an invaluable network with people from all over the world. I therefore thank all the WASP seniors for arranging courses, conferences and international study trips, and to all the WASP batch 1 students for the fun moments we shared.

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 your endless support and love. Lastly, I thank my precious wife Trang Nguyen and my son Minh Nguyen for standing beside me, especially my wife for the endurance, support, and unwavering love that will always be in my heart. I cannot thank you enough for bearing with me through the emotional ups and downs of my Ph.D. journey. My life is worth nothing without you.

Chanh Nguyen
Umeå, February 2021

Contents

Chapter 1

Introduction

1.1 Background and Research Motivation

Modern cloud platforms that can provide configurable computing resources (e.g., servers, storage, applications, and networks) on-demand with rapid provisioning and release have liberated application owners and companies from the need to plan resource provisioning far in advance and also relieved them of the burden of system administration and operation. In cloud computing platforms, cloud resources are packaged with a certain degree of abstraction in virtualized forms ranging such as virtual machines (VMs) and containers using specialized hardware and software. These cloud resources are typically centralized in a relatively small number of large datacenters located far from end-users. Since the cloud computing concept was first introduced and rigorously defined by the National Institute of Standards and Technology (NIST) in 2011 [MG11], many organizations have exploited its potential to increase IT efficiency and business agility by migrating and deploying many types of applications to the cloud, and by offloading applications or their components to cloud platforms. These applications include mobile interaction applications, parallel batch processing programs, data analytics tools, and extensions of compute-intensive desktop applications [Fox+09]. Most of them rely on highly available large datacenters to host large datasets and can be executed in parallel using hundreds of distinct compute resources. Such applications can tolerate moderate network delays and jitter on the order of hundreds of milliseconds, with tails of up to several seconds [Li+10]. They are therefore well suited to the *centralized deployment paradigm*. However, a new wave of emerging applications and services have much more stringent latency and jitter requirements.

More recently, the development of the Internet of Things (IoT) and advances in mobile technologies and artificial intelligence (AI) have resulted in the emergence of *a new wave of disruptive applications* for domains including health care [Isl+15], intelligent transportation [Bag+16], industrial process control [Jes+17], and everyday leisure [PM17]. Unlike traditional cloud applications, these new IoT applications are highly sensitive to network jitter and the variation in latency inherent to any multi-hop network. Moreover, they can generate vast quantities of data that are only of local interest but which require significant computational capabilities for processing and to ensure the necessary level of privacy [Elm+17]. Current application deployment mechanisms in which application-hosting nodes are located in centralized datacenters far from end-users are clearly not optimal for such applications.

For example, the quality and performance of state-of-the-art human-computer applications such as virtual reality (VR) and augmented reality (AR) applications or interactive online games are greatly reduced by high response times. A compelling AR system must support High Dynamic Range to ensure that the appearance of its virtual objects is spatially and temporally consistent with the real world, which requires a latency below 10 ms [Lin17]. These applications also require rapid processing of large volumes of data using complex technologies such as 3D rendering and machine vision. However, recent measurements indicate that typical latencies between end-users and public cloud datacenters are at least $20-40$ ms over high-quality wired networks, and up to $100-250$ ms over a 4G wireless network connection [Ska+18]. Obviously, these latencies are too high for such highly interactive human-computer applications to deliver instantaneous responses that appear natural to end-users. Accordingly, multiple studies have shown that current deployment strategies using centralized cloud infrastructures are sub-optimal for such applications because of the high latency and poor connectivity caused by long-distance communication [Che+17; Hu+16].

The same is true for edge content services such as YouTube Live, Facebook Live, and video surveillance systems, which generate large volumes of high definition video data (e.g., live streams of sporting events). Technologies that rely on centralized datacenters architecture to process and deliver content to millions of users are inefficient for such services for many reasons. First, forwarding such large amounts of data over the Internet to a centralized datacenter places considerable pressure on the core network. Second, in some cases the aggregated latency between the distant datacenter and the end-users may cause a poor quality of service. Third, there is a high risk of violating regulations on local network privacy policies due to a lack of location awareness and data privacy protection [KL10].

Large scale and industrial IoT systems (such as those used in smart cities or oil pipeline systems with millions of network-connected sensors) also generate vast streams of data that must be transferred to and processed by online analytics systems to enable real-time decision making. The International Data Corporation has predicted [SM20] that there will be around 41.6 billion connected IoT devices (e.g., cameras and sensors) by 2025, generating almost 79.4 zettabytes of data annually. Industrial IoT applications are expected to be responsible for a significant proportion of this. Most of the data generated by IoT devices is local in scope, i.e., it is needed only for local purposes such as coordinating the movements of self-driving cars at specific traffic hotspots,

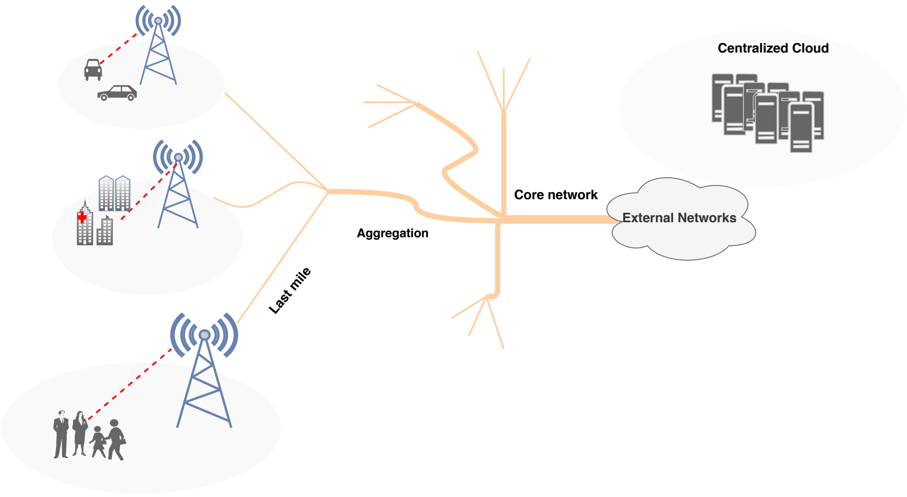

Figure 1.1: Different stage traversal for centralized cloud deployed applications' traffic.

evaluating gas transmission pipeline state information, or enabling intelligent control of an industrial process in a smart factory. It is therefore extremely costly and inefficient to host the data processing services on distant centralized nodes because doing so necessitates long-distance transfer of very large amounts of data that are only relevant in the local context. Additionally, transmitting such large quantities of data can easily cause congestion in the network if the aggregate demand exceeds the network capacity, and the network's latency and jitter may adversely affect the user experience or, worse, cause damage to people and/or the environment in the case of industrial process control systems.

The hype surrounding *fifth generation wireless network technology* (5G networks) has been building for years, and their potential has recently started to be realized. According to Ericsson [Jon+20], 15% of the world's population lived in an area with rolled out 5G coverage by the end of 2020. The worldwide number of subscriptions to 5G services is estimated to stand at 220 million as of the time of writing, and is expected to increase rapidly in the near future. 5G networks have several valuable attributes including *extremely high throughput* (up to 20 Gbps), *ultra-low latency* (round-trip latencies can be as low as few milliseconds from the end-user to the closest cellular base station), superior reliability, and massive network capacity [Ass15]. Consequently, they are likely to be central to the adoption and success of the emerging application ecosystem discussed above. However, the capabilities of 5G alone are insufficient to overcome the previously mentioned problems with deploying these emerging applications on centralized datacenters; indeed, the inclusion of 5G devices could potentially impose great pressure on network links if managed incorrectly. The key attributes of 5G, i.e.

its high coverage and bandwidth capacity, enable vast increases in the volume of data traffic originating from the network edge. Partly because of this, it has been predicted that approximately 75% of enterprise-generated data will originate from the network edge by 2025 [GR17]. Using current application deployment paradigms based on centralized datacenters, the traffic associated with cloud applications would have to traverse three different network parts, as shown in Figure 1.1:

- The *last-mile*: the first link from the end-user's premises to the outside world, via which the end-user directly transmits data to and receives data from their Internet Service Provider (ISP). The distance spanned by this layer is typically less than one mile.
- The *aggregation layer*: the link between the edge network and the point at which the ISP hands off the aggregated traffic to various network points of other providers.
- The *core network*: where off-premises or cloud datacenters are situated.

The *last-mile* is a vulnerable weak link because it can suffer congestion during peak hours depending on usage. This can be alleviated by adopting higher bandwidth network solutions such as 5G. Similarly, the *aggregation layer* and *core network* will suffer congestion if the aggregate demand exceeds their available bandwidth capacity. This slows down data transmission and significantly reduces users' QoE (quality of experience), especially for latency-intolerant applications. Unfortunately, modern telecom networks are too fragile to handle the enormous and rapidly varying capacity demands associated with next-generation cloud and IoT applications.

One way to mitigate the problems arising from the limited ingress bandwidth of centralized cloud infrastructures and improve the performance of web-based applications was introduced in early 1999, when Akamai developed *content delivery networks* (CDNs) to solve the *proximity problem* [Dil+02]. A CDN utilizes resources from servers deployed at the network edge in close physical proximity to end-users to cache static web content (e.g., images and documents). This improves applications' accessibility, reduces load times, and significantly reduces bandwidth consumption due to content replication. However, CDNs can only accelerate read-intensive applications such as those based on video streaming and web content.

The success of the CDN concept demonstrated the value of deploying certain applications, data, or application components in close proximity to their end users. However, for mobile and IoT applications that process data rather than merely transmitting it to the user, it is not sufficient to merely cache static content; instead, one needs a dispersed deployment of servers that are located in close proximity to end-users and data sources but which can also execute arbitrary code and perform data processing in the same way as would be done in a conventional centralized datacenter. Therefore, recent years have seen a transition towards a new type of computing infrastructure called *Mobile Edge Clouds* (MECs) in which substantial computing and storage resources are distributed at the edge of the network, close to end-users [Hu+15]. 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. In other words, MECs move processing capabilities and intelligence closer to end-users and data generating systems. Computation and processing at the network edge is achieved by exploiting the compute capacity of small servers or micro datacenters – referred to as Edge Data Centers (EDCs) – that are equipped or collocated with edge networking elements such as radio base stations or access points [Liu+18]. A range of terms have been used in the literature to describe concepts similar to MECs; examples include Cloudlets [Sat+13], Fog Computing [Bon+12; VR14], Mobile Edge Computing [Nun+15], and Telco Clouds [Bos+11; Soa+15]. These dispersed computing platforms were all designed to support future applications that require low latency and bandwidth scalability.

1.2 Characteristics of Mobile Edge Clouds

Figure 1.2 depicts a MEC system in which EDCs with heterogeneous scales and costs are distributed in close proximity to end-users in a wireless access network. In reality, MEC infrastructure may include tens of large datacenters and thousands of small ones collocated with cell towers and separated by distances of less than 10 miles [SBD18]. This allows MECs to provide computation and storage capabilities with higher bandwidth and lower latency than would be possible for a centralized cloud. MECs also offer other 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 [Liu+18]:

Ultra low latency. Because MEC resources are in close proximity to end users, applications can be deployed on EDCs, ignoring the rest of the network path to the distant cloud and therefore delivering low end-to-end application response times. With the capabilities of 5G networks, MECs can achieve extremely low latencies (on the order of several milliseconds). Additionally, network jitter between application-hosting nodes (i.e., EDCs) and end-users is minimized because the number of hops (i.e. transfers of data from one network segment to the next) is low.

Highly distributed and heterogeneous resources. The Edge Data Centers of MECs are distributed in different geographical locations within the wireless access network (i.e., at cellular base stations and access points).

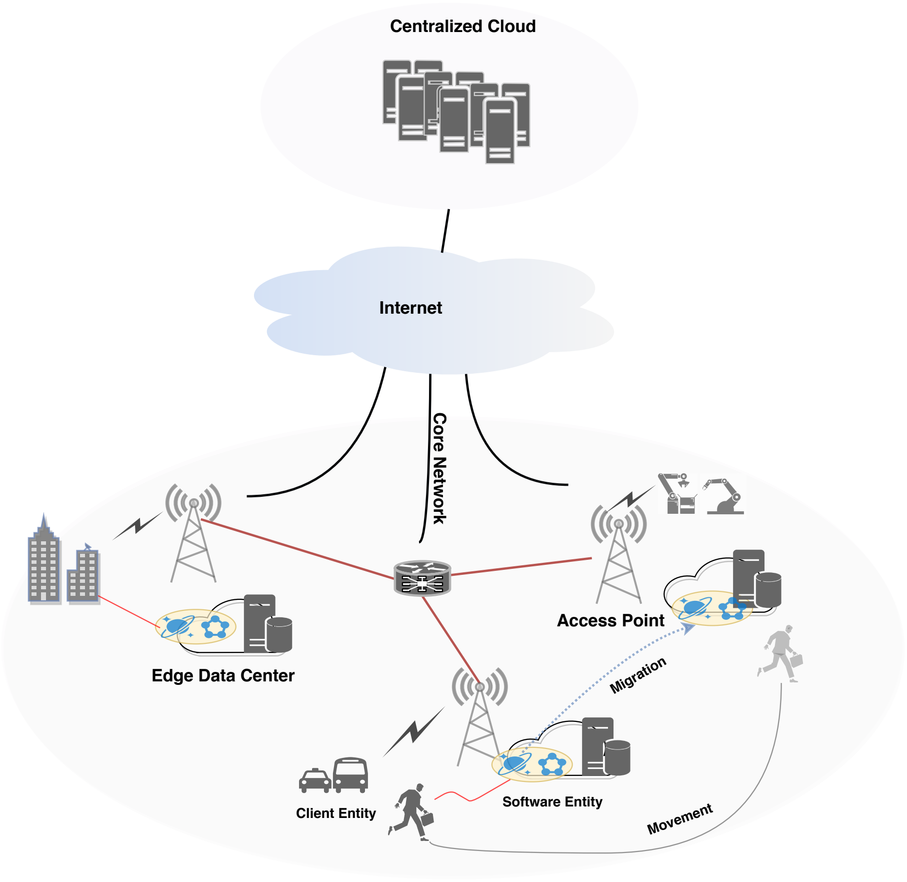

Figure 1.2: An illustrative MEC platform showing the association between client and service entities.

Furthermore, 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.

Support for mobility. MECs' clients are diverse; they include human beings with smartphones, IoT devices, sensors, and autonomous cars, among others. Therefore, the terms "end-user" and "client" can be used interchangeably in this context. All of them have a key common behavior, namely mobility, and are identified by a common attribute – location. In essence, they typically access MECs and often change their points of attachment to the network. Therefore, mobility support is critical for MECs.

Interplay with centralized clouds. MECs complement traditional centralized 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 also being less costly than MECs because they are located in more sparsely-populated areas with access to cheap electricity and cooling. Many applications and services may need to exploit the resources of both MECs and centralized clouds.

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 rather than conventional clouds 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 information about traffic at a hotspot to specific end-users close to that hotspot).

1.3 Research Problems and Objectives

MECs have been put forward as novel computing platforms that can overcome barriers to the success of future application types that are latency-sensitive, bandwidth-hungry, and compute-intensive. Experimental studies have validated and quantified the benefits of MECs, showing that considerable improvements in applications' end-to-end *response times* can be achieved by performing computation tasks at a nearby *EDC* rather than a distant cloud server. Processing large amounts of data at edge locations close to its point of origin greatly reduces *ingress bandwidth demand*. Additionally, edge servers located in the vicinity of end-users can serve as *privacy firewalls*, allowing end-users to dynamically and selectively control sensitive information from sensors [Sat+09]. In the decade since the MEC concept was first proposed, its importance and advantages have been widely acknowledged.

The development of MECs has been spurred by the potential power and utility of novel applications designed to benefit from their capabilities. Applications of this type generally will not perform satisfactorily if not deployed on MEC platforms. The success of such applications thus depends mutualistically on that of MECs, and we describe applications of this type as being *"MEC-native"*.

However, it is unrealistic to expect MECs to become successful based on these applications alone because MEC-native applications are unlikely to be developed extensively before MECs become widely available. MEC providers must therefore focus on the benefits MECs can offer to existing and widely used non-MEC-native applications. A promising class of applications that could 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**, is to answer the following two research questions:

1. Can cloud applications benefit from latency reduction when deployed on MECs?

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

Because MECs are designed to provide low-latency services and reduce network traffic to the central cloud, they basically offer 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 allocation. 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 a 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. Moreover, 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 change. The time taken to select and enact these resource allocation actions is important for two reasons: 1) resource usage is likely to vary rapidly in MECs, and 2) most applications deployed on MECs will be latency-intolerant, i.e. extremely sensitive to even very small delays. Sluggishness in resource scale-up or failure to allocate sufficient resources to meet demand can cause increased delays due to service waiting times, resulting in a bad user experience.

Given these challenges, MECs require autonomous resource allocation systems that can continuously monitor workload dynamics and adapt to changes by continuously optimizing resource allocations. The acquired resources must be transparently provisioned and ready to use so as to meet users' expectations. Consequently, the second research objective, **RO2**, is to understand the characteristics of MEC workloads and anticipate the likely variation in EDC workloads based on user mobility behavior. An efficient workload prediction model will help the resource management operator to pro-actively identify and make important management decisions. The research question pertaining to this objective is thus:

How can an efficient model for predicting MEC workloads be developed?

Elasticity is the ability of a system to automatically adapting resource provisioning to handle variation in load; it is a property that MECs must exhibit in order to become mature computing platforms. Within a given time window, a MEC attempts to provision resources such that the current demand is matched as closely as possible. However, achieving elasticity in a MEC is difficult for the reasons mentioned above. Therefore, the third research objective, **RO3**, is to develop methods and tools to help MECs overcome these challenges through automatic pro-active resource scaling. The research question associated with this objective is thus:

How can methods for efficiently auto-scaling MECs resources to meet the current demand be developed?

Answering this question made it possible to achieve objective **RO3** by developing an elasticity controller that allows MECs to pro-actively determine the proper amount of resources to allocate at each EDC.

Finally, because of the limited coverage area of base stations and the dynamic mobility of end-users, the problem of application placement in MECs is very challenging to solve, especially for stateful applications. As the user moves, the application should be migrated with them to ensure sufficient QoS and minimize bandwidth consumption due to the application's traffic. However, migrating too often may cause bandwidth wastage due to state migration traffic. The system must therefore decide in real-time where to place each application so as to minimize the total operating cost. To this end, the fourth research objective, **RO4**, is to address the following questions:

1. How can the workload, applications, and infrastructures be modeled? What are the various costs associated with operating applications on MECs?

2. Where should applications be placed among the available EDCs, and should an application be migrated as its user moves around so as to minimize the total operating cost?

The overall goal of **RO4** is to answer these questions in a practical manner by developing efficient real-time placement strategies for stateful applications in MECs.

To summarize, the main research objectives of this thesis are:

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

RO2 To develop workload prediction algorithms.

RO3 To develop an efficient elastic control framework for MECs.

RO4 To develop a real-time stateful application placement strategy for MECs.

1.4 Research Methodology

The research presented in this thesis was conducted using the constructive research method (CR) [Crn10] (also known as Design Science Research), which is a primary research method commonly used in the field of computer science. In essence, the output of a research project following this method is a solution that addresses a domain-specific practical problem, which is captured in an artifact such as an algorithm, model, or framework. The general process involves the following steps: 1) identifying a central research problem of practical relevance, as presented in Section 1.3; 2) obtaining a comprehensive understanding of the problem by undertaking a literature review, as presented in Chapter 2 and Chapter 3; 3) designing and developing applicable solutions to the problem such as the models, algorithms, and techniques to improve the efficiency of resource allocations in MECs, whose scientific contributions are presented in Section 1.5; and 4) demonstrating the efficiency and feasibility of the solutions, and linking the results obtained back to the research problem. In the works presented here, this step was accomplished by experimentally evaluating the performance of the proposed contributions and comparing them to alternative state-of-the-art solutions. Detailed information on the experimental setups and the approaches used for performance evaluation is presented in the experimental sections of each included paper.

1.5 Contributions of This Thesis

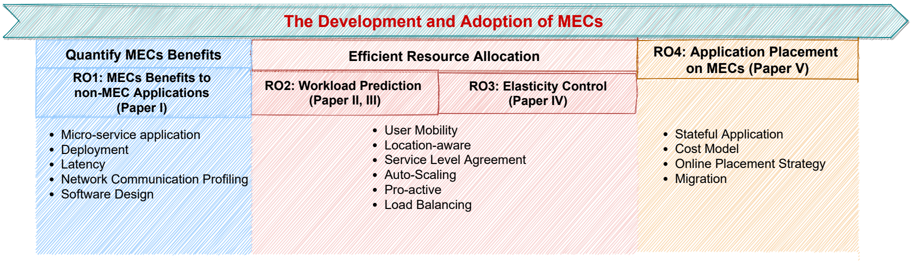
The Development and Adoption of MECs
Quantify MECs Benefits
RO1: MECs Benefits to non-MEC Applications (Paper I)
Micro-service application
Deployment
Latency
Network Communication Profiling
Software Design
Efficient Resource Allocation
RO2: Workload Prediction (Paper II, III)
RO3: Elasticity Control (Paper IV)
User Mobility
Location-aware
Service Level Agreement
Auto-Scaling
Pro-active
Load Balancing
RO4: Application Placement on MECs (Paper V)
Stateful Application
Cost Model
Online Placement Strategy
Migration

Figure 1.3: Thesis's main contribution.

MECs are still in their infancy, and their infrastructure configurations have yet to be standardized. As such there are many ongoing studies seeking to accelerate the adoption of MECs. This thesis contributes to these efforts in the following ways (as depicted in Figure 1.3):

- To spur MEC development, the first part of the thesis *extensively quantifies the benefits of MECs can offer to non-MEC-native-applications* (Paper I). One promising class of such applications are cloud-native applications, in particular *micro-service-based applications* with high deployment flexibility. We therefore *quantify the performance of cloud-native applications deployed using resources from both cloud datacenters and edge locations.* We then *develop a network communication profiling tool to identify the aspects of these applications* that reduce the benefits they derive from deployment on MECs. Finally, we *propose design improvements that would allow such applications to better exploit MECs' capabilities.*

- The second part of the thesis addresses problems relating to resource allocation in highly distributed MECs. First, to overcome the challenges arising from the dynamic nature of resource demand in MECs, we make use of statistical time series models (Paper II) and machine learning techniques (Paper III) to develop two *location-aware workload prediction models* for EDCs that account for both *user mobility* and *the correlation of workload changes among EDCs in close physical proximity*. Most existing approaches for workload prediction in the context of centralized clouds do not consider the impact of user mobility or information on the locations of datacenters. Therefore, they disregard potentially valuable inputs for improving the accuracy of workload prediction. Conversely, we leverage the cross-correlation of the workloads of nearby EDCs to achieve better predictions.

  The workload prediction model is then utilized to develop an *elasticity controller for MECs* (Paper IV). The proposed controller treats all EDCs located in close physical proximity as a group. Each group is managed by a group-level controller, which is responsible for three functions: 1) Predicting workload arrival at EDCs in the group; 2) Pro-actively determining how many resources to allocate at each EDC; and 3) Configuring load-balancers to direct requests from under-provisioned EDCs to EDCs within the group that have available resources. In essence, the elasticity controller helps MECs to perform resource allocation, i.e., to address the intertwined questions of what and how many resources to allocate, and when and where to deploy them.

- The third part of the thesis focuses on problems relating to *placement of stateful applications on MECs* (Paper V). This includes both questions of where to place applications, whether the requested applications should migrate with the user's movement so as to minimize the total operating cost while simultaneously guaranteeing sufficient end-user Quality of Service (QoS). These questions are easy to pose but intrinsically hard to answer due to the scale and complexity of MEC infrastructures and the stochasticity of user mobility behavior. To this end, we first *thoroughly model the workloads, applications, and infrastructures to be expected in MECs.* We then *formulate the various costs associated with operating the application*, namely the resource cost, migration cost, and service quality degradation cost. Based on our model, we *propose two efficient online application placement algorithms* that take these factors into account to minimize the total cost of operating the application.

The contributions of the thesis are described in more detail in Chapter 4.

1.6 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 briefly reviews the benefits of deployment on MECs for MEC-native applications, cloud-native applications, and legacy applications. Chapter 3 provides an overview of the main challenges and problems facing MECs, and presents a literature review focusing on previous efforts to solve the problems examined in this thesis.

Chapter 4 summarizes the contributions of the scientific works included in the thesis. Finally, chapter 5 briefly discusses ways of extending and building on the work presented here, and proposes some ideas for future research that could spur the development and adoption of MECs.

Chapter 2

Benefits of Mobile Edge Clouds for Various Application Types

It is clear that the wide geographical distribution of MECs' resources allows them to provide services with higher bandwidth and lower latency than current cloud computing platforms can deliver. As a result, MECs are promising platforms for hosting future applications with strong requirements for (extremely) low latency, high bandwidth, and strong computational capabilities. Such applications are referred to as MEC-native applications; it is expected that they will only perform optimally on MECs. Applications of this sort have recently attracted immense industrial and academic interest, spurring the development and adoption of MECs. However, it is unrealistic to expect MECs become successful based solely on MEC-native applications. Therefore, MEC providers and advocates should also focus on the benefits MECs can offer to non-MEC-native applications that are not specifically engineered for MECs but whose design and engineering are likely to make them suitable for deployment on MECs. Legacy applications and cloud micro-service applications are prime example of applications that might fall into this category. This chapter presents some illustrative examples from literature review of empirical studies on the potential benefits of deploying such applications on MECs.

2.1 MEC-native Applications

Some key demonstrators of the potential of MECs are latency-sensitive IoT-based applications such as *augmented reality and wearable cognitive assistance systems*. To evaluate the performance of such applications in terms of end-to-end response times, Zhou et al.[Che+17] conducted empirical experiments using 7 different cognitive assistance applications covering a wide variety of assistant tasks. Similarly, Hu et al.[Hu+16] examined the benefits of edge deployment in terms of response time and energy consumption by offloading different interactive mobile applications to the edge. Their results showed that *the performance of these applications was maximized by offloading to MECs rather than running on a centralized cloud.*

Similarly, Lin et al.[LS16] used resources in the vicinity of end-users to offload graphics rendering tasks for a Massively Multiplayer On-line Game (MMOG). This *reduced the remote cloud's ingress bandwidth consumption, increased user coverage, and reduced response latency.*

Real-time video analytics is a killer app of MECs, with uses ranging from face recognition in video surveillance systems to geo-distributed video analytics infrastructure for tracking traffic on roads. These systems generate large volumes of data that must be processed in real-time, and thus require considerable computational resources. Wang et al. [Wan+17] presented an open-source face recognition system combining video analytics with real-time denaturing for privacy. Using MEC resources, the system was able to maintain high accuracy while also achieving full frame rate speeds when deployed across a large number of cameras. Similarly, Mangiante et al. [Man+17] used resources from edge locations to enable 360° VR video streaming, yielding immediate *bandwidth savings.*

2.2 Cloud-native Applications

Modern cloud applications are increasingly architected as collections of micro-services [LF14]. The micro-service philosophy advocates constructing software applications as collections of small, independently deployable services that communicate with each other via light-weight mechanisms [New15]. An appealing property of micro-service-based cloud applications is their flexible deployment: they can be deployed in various configurations, combining resources in both centralized datacenters and edge locations. Figure 2.1 depicts a cloud application benchmark, namely Web Serving with three components deployed using resources from both an edge location and a centralized cloud datacenter.

In Paper I [Ngu+19], we quantified the benefits that an MEC can offer micro-service-based cloud applications by performing an empirical study using two popular benchmarks. The results obtained showed that deployment on an MEC did not significantly improve end-to-end latency even when most application services were deployed at the edge location because there were many transactions between application services when processing end-user requests. The number of transactions together with the network delay between the edge and the remote centralized cloud caused response times to increase dramatically. We proposed some ways to modify the engineering of cloud-native applications that could eliminate the adverse performance impact of latency between the

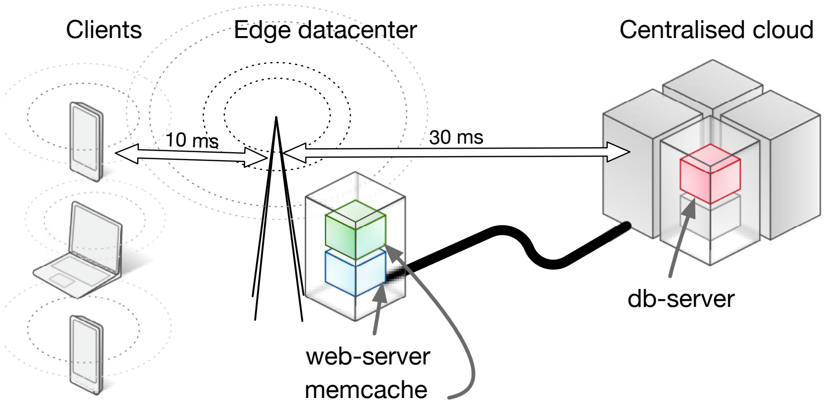

Figure 2.1: Illustration of an emulated MEC showing the network latencies between clients, an edge datacenter, and the centralized cloud with the Web Serving benchmark [Fer+12] deployed using resources from both an edge location and a centralized cloud.

edge location and the remote datacenter, thereby allowing the applications to benefit from deployment on MECs.

2.3 Legacy Applications

Legacy applications, for example those applications originally developed for personal computing environments that remain valuable today. Important examples include Adobe Photoshop and the Microsoft Office Suite. Mahadev et al.[Sat+20] investigated the potential for such *legacy applications to benefit from deployment on MECs* by introducing an edge-based virtual desktop infrastructure (EdgeVDI) that is used with customized virtual desktop infrastructures such as VMware or XenDesktop. EdgeVDI allows virtual desktops to be migrated on-demand from on-premise resources to an EDC close to a mobile user, ensuring the consistent low latency and high bandwidth required by remote desktop protocols. This approach allows end-users to access legacy applications on virtual desktops (deployed on nearby EDCs) using lightweight internet-connected devices with low computational power such as tablets or smartphones rather than having to carry comparatively heavy and cumbersome laptops.

The examples discussed above show that in addition to their proven benefits for MEC-native-applications, MECs could offer significant advantages for other type of applications. However, full exploitation of these advantages will require some design customization in the case of cloud micro-service applications, or middleware to support deployment on MECs in the case of legacy applications. Additionally, given the highly distributed and heterogeneous resource capability of MECs as well as the limited resources of EDCs, MECs need effective resource management operators that take application characteristics into account when making decisions about resource allocation.

Chapter 3

Resource Allocation in Mobile Edge Clouds

Reflecting the fact that MECs are currently in a very early stage of development, most studies on MECs have primarily focused on the concept of the MEC, its characteristics, and application scenarios that warrant further development of MECs [AA16; Por+18; You+19]. As discussed in Chapter 1, the characteristics of MECs differ from those of traditional clouds. However, they both have the same core challenge – how to efficiently provide resources (computation, bandwidth, and storage capabilities) and services to end-users while fulfilling the objectives of both infrastructure providers and application providers, which typically include maximizing energy efficiency, achieving service level objectives, and optimizing operating costs, among others [BB10; You+10; Che+18].

To promote the development of MECs and their maturation as computing platforms, this thesis investigates methods and approaches for helping MECs to manage resources and services efficiently. Accordingly, this chapter begins by describe the challenges associated with resource allocation in MECs. It is critical to understand the stakeholders who directly impact and are impacted by the resource management decisions in MECs. Therefore, we also identify key MEC stakeholders and the goals (i.e., core metrics) that each stakeholder aims to achieve. We then discuss the modeling of the key components and performance-affecting factors of an MEC – its infrastructure, the applications it hosts and their workload, and its end-users. Finally, we present a literature review highlighting state-of-the-art studies that have sought to address these challenges by developing resource allocation mechanisms for MECs.

3.1 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. This is beneficial because it reduces latency and mitigates congestion in the network core, unlocking the potential of new application types including IoT applications, autonomous vehicle systems, and AR/VR applications [PM17; Mah+18; Shi+16]. 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 datacenters of various sizes collocated 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 each end-user with resources from EDCs located in their 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. Modern 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.

3.2 Stakeholders of MECs and Their Goals

Resource allocation is the process of allocating, scheduling, and planning resources to maximize resource usage efficiency while guaranteeing that the predefined service level objectives are met. In the MEC ecosystem, the *in-*

frastructure provider and *end-user* are stakeholders who both impact and are impacted by resource allocation decisions.

- The main goals of the infrastructure provider are to maximize average resource utilization and maintain system stability.
- The main goal of end-user is to be served with the greatest possible quality of the service.

Therefore, a resource management operator must guarantee that all decisions concerning resource allocation and provisioning take these core metrics into account. Unfortunately, these metrics may be in conflict under certain conditions. For example, increasing resource utilization may increase the rejection ratio, resulting in a poor quality of experience for users. Therefore, an ideal management operator should strike a balance between these metrics and adjust their prioritization depending on the situation at hand. For example, the resource allocation for latency-intolerant mission-critical applications may prioritize reducing the rejection ratio over maximizing resource utilization.

The performance of resource allocation tools and approaches can be meaningfully and quantitatively evaluated using a set of system- and user-oriented metrics recommended by the Standard Performance Evaluation Corporation (SPEC) [Her+16]. These metrics include: *under-* and *over-provisioning accuracy*, which measure the deviation between resource demand and resource supply; *under-* and *over-provisioning timeshare*, which measure the proportion of the total time during which the system is under- or over-provisioned; and *instability*, which indicates whether the supply curves change in the same direction as the demand curve.

3.3 Modeling MEC Components

The primary inputs that an MEC resource management operator requires to make decisions about management actions are information on the key components of the MEC system, namely the infrastructure of the MEC, the applications it runs and their workloads, and its users. Figure 3.1 shows the system components relevant to resource management in MECs.

3.3.1 Infrastructure Modeling

Academic and industrial groups have attempted to develop unified standards for MEC infrastructure [CPS17; Yu16]. It is very likely that most MEC infrastructures will have a hierarchical tree topology similar to that of mobile core and access networks [Bed14]. The resources of a distributed MEC are the compute and storage capabilities of its datacenters and the bandwidth capacity and throughput of the network links connecting these datacenters. From the MEC infrastructure provider's perspective, a key resource management objective is to minimize the total operating cost of providing services. It is therefore

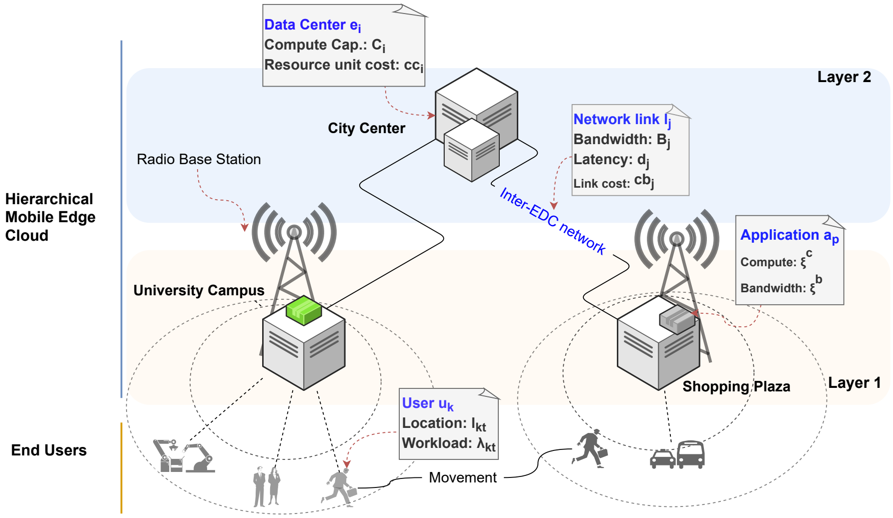

Figure 3.1: The primary components relevant to resource allocations in MECs.

essential to develop robust cost models for these resources. These cost models must capture the heterogeneity of MEC resources, which can be expressed in terms of generalized measures of capacity and capability such as compute and bandwidth units [Meh+16; Tär+17]. Cost models can also be dynamic, allowing them to take into account the law of supply and demand [Moo25] as well as the impact of economy-of-scale effects on energy and maintenance costs [NKE].

In our work, we model MEC infrastructures in various ways depending on the research objective in focus. For example, when investigating the relationship between neighboring EDCs (location-awareness), we model a MEC as a hexagonal grid with each EDC located in each cell that is distributed over an area [LKE17] and also examine a hypothetical MEC whose EDCs are collocated with cellular towers in their real geographical positions within a real-world landscape (e.g., the San Francisco bay area or Rome) [NKE19; NKE20]. Conversely, when investigating the application placement problem, we model a MEC as a hierarchical tree architecture [NKE] with EDCs having different resource capacities and costs.

3.3.2 Application and Workload Modeling

A wide range of different applications with differing resource requirements could be deployed on MECs to leverage their resources. For example, a deployed application could be a single component service or a multi-component application in which each component has distinct resource requirements. Additionally, the application could either be stateless or stateful; in the latter case, it will be necessary to manage its state when deploying the application or migrating it over the MEC's resources.

An application model must capture the application's resource demand, including the demand of each individual component in the case of a multi-component application as well as the demand associated with state data for stateful applications. Resource demand can be quantified as the *average resource requirement over time unit per request* with respect to the consumed resource types. Depending on the nature of the application, each request may consume a large amount of compute resources compared to bandwidth resource or vice versa. The nature of the application in this respect can be defined based on its *compute-intensity-to-bandwidth-usage-ratio* [Meh+16]. The more detailed the application model, the better the understanding of how the provisioning of resources and capacity in response to its requests will affect its key performance indicator (e.g., the expected application response time) at any given time.

Our application workload models allow for variation in the number of requests both in time and between locations [NKE20]. This is especially important in MECs due to the stochasticity of user mobility and the wide distribution of MEC resources.

3.3.3 User Modeling

As mentioned in Chapter 1, MECs users (or subscribers) may be human beings, IoT devices, or cameras, among other things. These users are characterized by high mobility, so their locations will change periodically. The stochastic nature of mobility is a major challenge when deciding how to allocate resources at EDCs. It is therefore desirable to model the variation of users' locations and requested services over time. Because the development of MECs is still in its early stages, there are no publicly available datasets of MEC end-user behavior and workloads. Therefore, many studies rely on real-world user mobility traces to simulate synthetic user behavior and workload in MECs [BG20; Urg+15]. Accordingly, we use the mobility traces of taxis in specific real-world locations (Rome and the San Francisco Bay area) to simulate user mobility. The main reason we chose these taxi traces is that they cover the same geographic area as the data used to generate the geographic distribution of our emulated MEC infrastructures.

3.4 A Literature Review on Resource Allocation in Mobile Edge Clouds

The main goal for a mature computing platform is to ensure that services are provided with the greatest possible reliability and availability while meeting performance targets and minimizing costs and energy consumption. In MECs, the challenges mentioned above make traditional centralized resource management strategies that rely on human intervention impractical [Tär+15; AS07]. It will

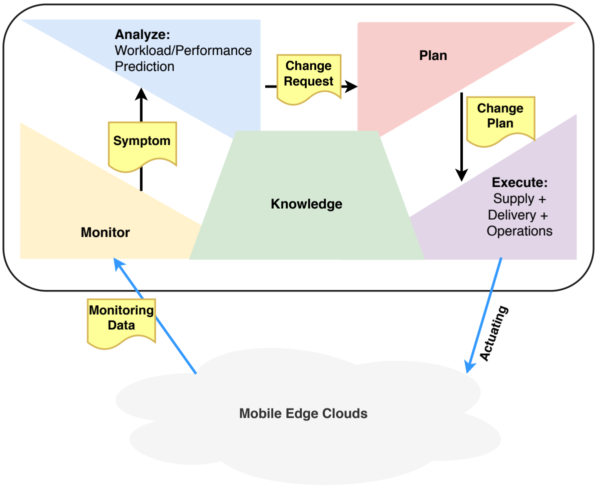

Figure 3.2: An MAPE-K loop resource management system in a MEC.

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 essence, an autonomic system is a system with a hierarchy of self-governing components, each consisting of multiple interacting autonomous components [KC03]. For example, Figure 3.2 shows a MAPE-K loop-based autonomic resource management system for a MEC. Here, the MEC resource management system continuously adapts the resource allocation and provision behaviour to achieve predefined goals using an intelligent loop with 5 components: monitor (M), analyze (A), plan (P), execute (E), and knowledge (K).

- The *monitor* component periodically gathers different metrics relating to the MEC 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.

- The *analyze* component applies different complex data analysis mechanisms such as statistical models, time series models, and machine learning techniques to capture the static and dynamic characteristics of the MEC's resources and the hosted applications, as well as the behavior of the real workload that MEC processes.

- The *plan* component is responsible for planning mitigation actions that will allow the MEC to adapt to predicted changes. Using results generated by the *analyze* component together with predefined target performance indicators relating to variables such as throughput and response times, the autonomic manager structure actions (e.g., admission control, resource allocation, migrations) to ensure the MEC meets its performance target while minimizing costs and energy consumption.
- 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.
- 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 the MEC. Similarly, the *execute* component might update the knowledge base with records of action taken in response to the output of the *analysis* and *plan* components, making it possible to trace the actions' effects on the system.

While the fundamental principles of autonomic systems are relatively well understood, the extreme scale, complexity, and dynamics of MECs makes the practical implementation of those principles very difficult [Car+18].

This section reviews the efforts that have been made to address these challenges and develop solutions that facilitate efficient resource allocations in MECs. We first consider research on *Workload Prediction*, which is essential for any management operator. Understanding and accurately predicting how a system's workload will change can improve the quality of resource management decisions in MECs. We then examine 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. Finally, 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 pool of a MEC to ensure that the required Quality of Service is delivered while minimizing operating costs.

3.4.1 Workload Prediction

Understanding and modeling workload behavior is essential for efficient management of cloud system resources. Consequently, many published studies have focused on modeling and predicting workloads in cloud datacenters.

Many researchers have proposed workload prediction models that use different classical statistical models (e.g., Markov models, Bayesian models, or time series), or machine learning techniques such as artificial neural networks or deep learning. For example, Khan et al. [Kha+12] clustered repeated workload patterns among VMs into different groups, then used a Hidden Markov Model to explore temporal correlations and variation in workload patterns. Similarly, Sheng et al. [DKC12] used a Bayesian model to perform both short- and long-term mean load prediction. On the basis of experiments using real traces from Google, they claimed that the proposed method outperform alternatives based on time series and filters. Approaching the problem from another perspective, Kumar et al.[KS18] used a neural network and a differential evolution algorithm to develop a workload prediction tool for cloud datacenters. This model is capable of learning and extracting workload patterns, and achieves substantially lower prediction errors than alternative models. Similarly, Zhang et al.[Zha+18b] built a deep learning model based on the canonical decomposition to predict cloud workloads. The proposed model achieves a better performance when performing with the complex workload data.

The heterogeneity of MECs and the complexity of their workloads makes it difficult to fully capture the characteristics of their workloads using a single predictive tool. Therefore, some researchers have proposed hybrid approaches that combine multiple tools. For example, Chen et al.[Che+15] proposed an ensemble prediction model that uses multiple base predictors and a fuzzy neural network to improve predictive accuracy. Unfortunately, the complexity of this model means that it must perform multiple computational steps per prediction interval, preventing its use in real-time systems. Conversely, Liu et al. [Liu+17] presented an adaptive categorical workload prediction framework that categorizes workloads based on their characteristics (e.g., the speed of workload change and the priority of the jobs) and uses a different predictive model for each workload category. Experiments using workload traces from a real cloud showed that the proposed model outperformed alternative time series-based predictive methods.

User mobility and the wide geographic distribution of EDCs in MECs present new challenges in workload prediction. Unfortunately, the techniques mentioned above only take into consideration information on individual server or application workloads, and therefore may not work efficiently in the context of MECs. In Paper II [LKE17] and Paper III [NKE19], we proposed two location-aware workload prediction tools for EDCs that use a vector autoregressive model and a multivariate long short-term memory network, respectively. The proposed models exploit the correlation of workloads between nearby EDCs to forecast the future workload of each EDC, and were shown to outperform alternative state-of-the-art methods.

3.4.2 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 questions of what and how many resources to allocate to each hosted application are not trivially answered. This is especially true for MECs, whose infrastructure makes this challenge much more severe than 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 [YLL18; Che+18] 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 [YLL18]. This tool uses the delay constraints of the managed tasks to schedule and allocate resources from edge nodes or a centralized data center based on the objective of ensuring that the response times of the managed tasks remain below predefined thresholds. Chen et al. [Che+18] 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 ensures satisfaction of performance guarantees for the managed applications.

Another effort to address the capacity sizing problem was presented by Mehta et al. [Meh+18], who developed a two-tier scheduler for allocating runtime resources to industrial IoT 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 [Car+18] 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.

Most applications deployed on MECs will be latency-intolerant and extremely sensitive to small delays. Furthermore, due to the limited availability of compute resources at the network edge, the resource costs are expected to be more expensive than that of centralized cloud. Therefore, allocating resources exceeding the demand leads to inefficient operation and costly resource wastage. All in all, resource allocation in MECs must be more rigorous in terms of speed and precision than those in centralized cloud datacenters. With this in mind, Paper IV [NKE20] presents an elasticity controller that helps MECs to automatically adapt resource provisioning to handle variation in the arrival workload. We used queueing theory techniques to build a performance model that estimates the number of resources that should be provisioned to EDCs in order to meet predefined Service Level Objectives (SLOs) while maximizing resource utilization. The controller also incorporates a group-level load balancer that is responsible for redirecting requests among EDCs during runtime so as to minimize the request rejection rate.

3.4.3 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 centralized data centers to edge data centers) 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 data center, while the face recognition component could be deployed on the centralized distant data center. 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 data center or cloud system; in the case of an MEC, each component of a cloud application could be placed anywhere from a centralized distant data center to an EDC near the user. The service placement problem in MECs is complicated by several factors that do not affect conventional clouds, including the limited coverage area of base stations, the dynamic nature of mobile users, and network background traffic. Nevertheless, this problem 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. [TLG16] 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. [Tär+17] 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. [Wan+18] introduced ITerative Expansion Moves (ITEM) to solve the combinatorial optimization problem for service entity placement. In another notable study [WZL17], Wang et al. modeled users, a multi-component application, and physical

MEC resources as graphs and considered service placement based on a linear application graph with the goal of minimizing peak resource utilization for both compute resources and network links. To achieve this goal, the authors proposed online approximation algorithms for lacing tree application graphs onto tree physical graphs. Taking into account stochastic user mobility, Ouyang et al. [OZC18; Var+19] proposed efficient heuristic algorithms to optimize long-term time-averaged migration costs. The same research group subsequently proposed a novel mobility-aware online service placement framework to achieve a desirable balance between user latency and migration cost [OZC18], as well as a joint service placement and routing algorithm designed to minimize total service placement costs [Var+19].

We realize that most of the studies on application placement in MECs consider only *stateless applications*. However, *many envisioned MEC applications are stateful.* Technically, a stateful application has a user state (or application state) to store the context and history of the previous transaction so that the next transaction can perform with the context of previous transactions. For example, artificial reality applications must store generated meshes, world data, generated textures, etc. This stateful architecture causes more challenges for MECs to decide where to place such applications. Employing stateless placement algorithms for stateful applications risks introducing unnecessary costs due to wastage of the bandwidth required to migrate user state from one EDC to another. To this end, in Paper V [NKE], we address the problem of placing stateful applications in MECs. First, we thoroughly model the workloads, applications, and infrastructures to be expected in MECs. We then formulate the various costs associated with operating an application, namely resource cost, migration cost, and service quality degradation cost. Finally, we propose two efficient online placement algorithms: *Follow-me* and *Gale-Shapley*-based algorithm. Our experimental results show that both of these algorithms can help MECs rapidly decide where to allocate capacity for applications, achieving total operating costs that are no more than 8% higher than the approximate global optimal.

Chapter 4

Summary of Contributions

This chapter summarizes the papers comprising this thesis and shows how they relate to the targeted research objectives. First we present an overall outline of the contributions of this work. This is followed by more detailed discussions of the five included papers in chronological order, with descriptions of the author's contributions.

4.1 Outline of Contributions

This thesis focuses on four research objectives that were addressed in three parts. 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 and autonomous vehicles). 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 must also 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 part of this thesis addresses problems relating to resource allocations in MECs. Although there have been many studies on workload modeling and prediction in the context of cloud datacenters, there remains a lack of reliable tools for workload prediction in MECs. The wide distribution of EDCs and user mobility behavior present new challenges for workload prediction in MECs. Because state-of-the-art workload prediction techniques only take into consideration knowledge of individual server or application workloads, they may not work efficiently in the context of MECs. We therefore used statistical time series models and machine learning techniques to develop efficient workload prediction models that take these factors into account. Papers II and III introduce two location-aware workload prediction models for EDCs that account for both user mobility and the correlation of workload changes among EDCs in close physical proximity. Paper IV describes the use of these workload prediction models in an elasticity controller for MECs that was developed to manage resource allocation for latency-sensitive applications.

The third part of the thesis addresses the problem of placing stateful applications in MECs. To ensure optimal QoS while minimizing bandwidth consumption due to application traffic, applications should be migrated in parallel with the movements of their users. However, migrating too often may cause bandwidth wastage due to state migration traffic. To address this problem, we first thoroughly model the workloads, applications, and infrastructures to be expected in MECs. We then formulate the various costs associated with operating the application, namely resource cost, migration cost, and service quality degradation cost. Finally, we propose two efficient online placement algorithms which can help MECs rapidly decide where to place allocate capacity for applications, achieving total operating costs that close to the approximate global optimal.

In the following sections, we present a summary of each paper.

4.2 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'19), pp. 250-263, ACM, 2019.*

Paper Contributions

In Paper I [Ngu+19], 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 micro-service-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.

Authors Contributions

I was the main author who contributed to the formulation of the problem, conducted the experiments, and wrote all the main parts of the paper. Amardeep Mehta helped design the Web Serving experiments and add paragraphs regarding 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.

4.3 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.*

Paper Contributions

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 [LKE17], 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 decrease 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.

Authors Contributions

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

4.4 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 (CCGrid), pp. 341-350, IEEE/ACM, 2019.*

Paper Contributions

Paper III [NKE19] also addresses part of **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 shaking techniques clearly demonstrated that the neural network-based method is stable and robust.

Authors Contributions

I was the main author who contributed to the problem formulation, 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.

4.5 Paper IV

Chanh Nguyen, Cristian Klein, and Erik Elmroth. **Elasticity Control for Latency-Intolerance Mobile Edge Applications**. *In Proceedings of the 5th ACM/IEEE Symposium on Edge Computing (SEC'20), pp. 70-83, ACM, 2020.*

Paper Contributions

Elasticity is a key property required for MECs in order to become mature computing platforms hosting software applications. It is the ability to automatically adapt resource provisioning as required to handle variation in load. In MECs, the elastic resource allocation controller must be even more rigorous in terms of speed and precision than those in centralized cloud infrastructures for the following main reasons: 1) Most application deployed on MECs are latency-sensitive, which is sensitive to even very small delays. Sluggishness in resource scale-up or failure to allocate sufficient resources to meet demand (i.e., under-provisioning) can cause delays by increasing service waiting time, results in a bad user experience; 2) The limited availability and high cost of resources at the network edge mean that allocating resources exceeding the demand (i.e., over-provisioning) leads to inefficient operation and costly resource wastage; 3) The stochastic nature of user mobility means that resource demand at the network edge is characterized by frequent transient changes.

In Paper IV [NKE20], we address **RO3** by proposing a location-aware elastic controller for MECs. The proposed controller takes advantage of the correlation of workload variation in physically neighboring EDCs to predict the request arrival rate at EDCs. These predictions are then used as inputs to estimate service demand and the number of resources that will be desired at each EDC. Additionally, to minimize the request rejection rate, we develop a group-level load balancer to redirect requests among EDCs during run-time.

We evaluate the performance of the proposed using various core elasticity metrics (as presented in Chapter 3). Experiments using an emulated MEC over a metropolitan area (San Francisco area), and simulated application workloads from a real mobility trace (San Francisco taxi trace) show that the proposed controller delivers significantly better scaling behavior than a state-of-the-art re-active controller and also improves the efficiency of resource provisioning. The proposed elastic controller helps MECs maintain resource utilization and request rejection rates that satisfy predefined requirements while maintaining system stability.

Authors Contributions

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.

4.6 Paper V

Chanh Nguyen, Cristian Klein, and Erik Elmroth. **State-aware Application Placement in Mobile Edge Clouds**. *Submitted for journal publication, 2021.*

Paper Contribution

Many envisioned MEC applications are stateful. Placement of such stateful applications on MECs is challenging due to the stochastic nature of user mobility. Employing stateless placement algorithms for stateful applications risks introducing unnecessary costs due to wastage of the bandwidth required to migrate user state from one EDC to another.

In Paper V [NKE], we address **RO4** by proposing two online state-aware application placement algorithms for MECs, named *Follow-me* and *Gale-Shapley*-based algorithms. We start by thoroughly modeling the costs incurred by stateful applications on MECs, namely the resource cost (consisting of computing cost and application bandwidth cost), QoS degradation cost, and migration cost. The two proposed online placement algorithms aim to minimize the total operating cost, i.e. the sum of these three individual costs.

We evaluate these proposed algorithms using an MEC topology consisting of base stations geographically distributed across the San Francisco area. User mobility is modeled using real mobility traces of taxis in San Francisco. Finally, users' transition between applications are modeled based on a Markov model. Our results show that the two proposed online placement algorithms can efficiently decide where to place applications among EDCs, reaching a total operation cost only 8% below the approximate global optimal placement provided by the clairvoyant offline algorithm. Of the two online algorithms, the Gale-Shapley-based algorithm achieves better optimal solutions than the Follow-me algorithm, reducing operating costs by up to 17% while helping MECs to effectively balance workloads to mitigate resource scarcity.

Authors Contribution

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.

4.7 Limitations

MECs still in their infancy, and their infrastructure configuration has yet to be standardized. Therefore, the work presented in this thesis primarily involved experiments on emulated and simulated systems, which made it necessary to apply some simplifying assumptions. Based on the research problems under investigation, we chose to exclude some aspects of real MECs that were expected to present complications.

Because the main objective of the study presented in Paper I was to quantify the latency reduction achieved when deploying existing applications using MEC resources, we configured an emulated MEC and studies the network delays between end-users, edge locations, and a distant centralized datacenter.

In Papers II, III, and IV, we focused on the correlation of workload changes in EDCs located in close physical proximity to one-another. To this end, we emulated MECs with multiple topologies – one in which the EDCs were distributed over an area with a hexagonal topology (Paper II) and another in which the distribution of EDCs was based on the real-world geographical distribution of cellular base stations in San Francisco (Papers II and IV).

Paper V examined an MEC with a hierarchical topology in which the EDCs in the lowest layer were collocated with real cellular base stations in San Francisco.

These assumptions may limit the direct applicability of the results presented here in certain real world scenarios. Consequently, the developed tools and algorithms require further testing and may need to be extended based on the outcomes of that testing.

Chapter 5

Future Research Directions

In this thesis, we propose techniques and methods to address different fundamental resource management challenges associated with MECs, including workload modeling and prediction; resource provisioning and allocation; and workload and application placement. The works pertaining to this thesis is expected to continuously evolve along with the development of Mobile Edge Clouds platforms, where the challenges and issues continuously evolve and need further investigations. In spite of the significant contributions of the current thesis, there are many open research challenges that need to be addressed in order to further advance the area. This chapter outlines several open issues that are promising unexplored pathways for future research.

5.1 Decentralized Control Plane

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 and do not scale well with the number of users and applications or the size of the infrastructure. Decentralized autonomic strategies are thus preferable. A better approach is to design a *decentralized control plane* with a local controller at each edge cloud location that manages local resources within the cluster. Because many edge cloud orchestration tasks require at least partial information from controllers in their vicinity, a robust way to share information among these controllers is needed.

5.2 Incorporating Last-mile Bandwidth

Servers in traditional clouds are typically connected via very high speed networks – typical network speeds in datacenters range from 10 Gbps to 100 Gbps, giving the hosted cloud applications abundant network resources. Conversely, in MECs, users will share the air interface by multiplexing over a limited set of frequencies. Connection bandwidth is thus likely to be a bottleneck. *Incorporating network bandwidth considerations into scaling and placement decisions* is thus another key issue that must be addressed.

5.3 Application Development for MECs

The historical development and adoption of traditional clouds shows that it was important for existing applications to benefit from cloud deployment, we conclude in [Ngu+19] that: "*without applications benefiting from clouds, cloud providers would have been reluctant to invest in infrastructure, and the lack of cloud infrastructure would have made application providers reluctant to develop for the cloud*". Therefore, to increase momentum towards MEC adoption and development, it will be necessary to investigate the software architectures needed to develop and customize applications that perform well when deployed on MECs. An *MEC programming model* that simplify developing of geo-spatially distributed, large-scale, and latency-sensitive applications is necessary to investigate [Hon+13; Ha+14].

5.4 Multi-tenant MECs

The resources of MECs are virtualized and allocated to multiple users simultaneously. However, there is a lack of studies on multi-tenant support in MECs in the current literature. For example, there is a need to investigate ways of efficiently scheduling multiple tasks and applications on MECs' resources while taking their SLOs into account.

5.5 Energy-efficient MECs

Sustainability will be essential for the realization and acceptance of MECs as a future computing platform. Modern centralized data centers consume a lot of energy, emit a lot of carbon dioxide, and generate significant electronic waste [KÅN20]. There has been little research on optimizing energy usage in MECs, but their sustainability could potentially be improved by investigating techniques for consolidating EDCs by migrating tasks/applications from one EDC to another. In addition, optimal strategies for task migration must be developed.

5.6 Trustworthiness of MECs

The distributed nature of MECs will probably give them a larger attack surface than centralized cloud systems. Therefore, building robust MECs that can remain functional in the presence of malicious attacks is essential. Further, EDCs will likely function in part as distributed storage systems for local data. Therefore, it is important to develop tools that ensure the security of data sources and preserve the user privacy at the edge [Zha+18a; Wan+19].

Bibliography

[AA16] 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.

[AS07] 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.

[Ass15] 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 Oct. 1, 2020).

[Bag+16] 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.

[BB10] Anton Beloglazov and Rajkumar Buyya. "Energy efficient resource management in virtualized cloud data centers". In: *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE. 2010, pp. 826–831. ISBN: 978-1-4244-6988-8. DOI: 10.1109/CCGRID.2010.46.

[Bed14] Paul Bedell. *Cellular Networks: Design and Operation: A Real World Perspective*. Outskirts Press, 2014. ISBN: 978-1478732082.

[BG20] Tayebeh Bahreini and Daniel Grosu. "Efficient Algorithms for Multi-Component Application Placement in Mobile Edge Computing". In: *IEEE Transactions on Cloud Computing* (2020). DOI: 10.1109/TCC.2020.3038626.

[Bon+12] 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.

[Bos+11] Peter Bosch, Alessandro Duminuco, Fabio Pianese, and Thomas L Wood. "Telco clouds and virtual telco: Consolidation, convergence, and beyond". In: *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. IEEE. 2011, pp. 982–988. DOI: 10.1109/INM.2011.5990511.

[Car+18] 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.

[Che+15] Zhijia Chen, Yuanchang Zhu, Yanqiang Di, and Shaochong Feng. "Self-adaptive prediction of cloud resource demands using ensemble model and subtractive-fuzzy clustering based fuzzy neural network". In: *Computational intelligence and neuroscience* 2015 (2015).

[Che+17] 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.

[Che+18] 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.

[CPS17] Alberto Ceselli, Marco Premoli, and Stefano Secci. "Mobile edge cloud network design optimization". In: *IEEE/ACM Transactions on Networking* 25.3 (2017), pp. 1818–1831. DOI: 10.1109/TNET.2017.2652850.

[Crn10] Gordana Dodig Crnkovic. "Constructive research and info-computational knowledge generation". In: *Model-Based Reasoning in Science and Technology*. Springer, 2010, pp. 359–380. DOI: 10.1007/978-3-642-15223-8_20.

[Dil+02] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. “Globally distributed content delivery”. In: *IEEE Internet Computing* 6.5 (2002), pp. 50–58. ISSN: 1941-0131. DOI: 10.1109/MIC.2002.1036038.

[DKC12] Sheng Di, Derrick Kondo, and Walfredo Cirne. “Host Load Prediction in a Google Compute Cloud with a Bayesian Model”. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. SC ’12. Salt Lake City, Utah: IEEE Computer Society Press, 2012, 21:1–21:11. ISBN: 978-1-4673-0804-5. URL: http://dl.acm.org/citation.cfm?id=2388996.2389025.

[Elm+17] Erik Elmroth, Philipp Leitner, Stefan Schulte, and Srikumar Venugopal. “Connecting fog and cloud computing”. In: *IEEE Cloud Computing* 4.2 (2017), pp. 22–25. ISSN: 2325-6095. DOI: 10.1109/MCC.2017.29.

[Fer+12] Michael Ferdman, Almutaz Adileh, Onur Kocberber, Stavros Volos, Mohammad Alisafaee, Djordje Jevdjic, Cansu Kaynak, Adrian Daniel Popescu, Anastasia Ailamaki, and Babak Falsafi. “Clearing the clouds: a study of emerging scale-out workloads on modern hardware”. In: *Acm sigplan notices* 47.4 (2012), pp. 37–48. DOI: 10.1145/2248487.2150982.

[Fox+09] Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andrew Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. “Above the clouds: A berkeley view of cloud computing”. In: *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS* 28.13 (2009).

[GR17] Bob Gill and Santhosh Rao. *Technology Insight: Edge Computing in Support of the Internet of Things*. July 2017.

[Ha+14] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. “Towards Wearable Cognitive Assistance”. In: MobiSys ’14. Bretton Woods, New Hampshire, USA: Association for Computing Machinery, 2014, pp. 68–81. ISBN: 9781450327930. DOI: 10.1145/2594368.2594383.

[Her+16] Nikolas Herbst, Rouven Krebs, Giorgos Oikonomou, George Kousiouris, Athanasia Evangelinou, Alexandru Iosup, and Samuel Kounev. “Ready for rain? a view from spec research on the future of cloud metrics”. In: *arXiv preprint arXiv:1604.03470* (2016).

[Hon+13] Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. “Mobile fog: A programming model for large-scale applications on the internet of things”. In: *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. 2013, pp. 15–20. DOI: 10.1145/2491266.2491270.

[Hu+15] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. “Mobile edge computing—A key technology towards 5G”. In: *ETSI white paper* 11.11 (2015), pp. 1–16.

[Hu+16] 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.

[Isl+15] 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.

[Jes+17] 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.

[Jon+20] Peter Jonsson, Steven Davis, Peter Linder, Amir Gomroki, Ali Zaidi, Anders Carlsson P, Miljenko Opsenica, Ida Sorlie, Sebastian Elmgren, Greger Blennerud, Harald Baur, Ritva Sveningsson, and Brian Heath. *Ericsson Mobility Report*. Tech. rep. Nov. 2020.

[KÅN20] Carolina Koronen, Max Åhman, and Lars J Nilsson. “Data centres in future European energy systems—energy efficiency, integration and policy”. In: *Energy Efficiency* 13.1 (2020), pp. 129–144. DOI: 10.1007/s12053-019-09833-8.

[KC03] 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.

[Kha+12] Arijit Khan, Xifeng Yan, Shu Tao, and Nikos Anerousis. “Workload characterization and prediction in the cloud: A multiple time series approach”. In: *2012 IEEE Network Operations and Management Symposium*. Apr. 2012, pp. 1287–1294. DOI: 10.1109/NOMS.2012.6212065.

[KL10] 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.

[KS18] Jitendra Kumar and Ashutosh Kumar Singh. "Workload prediction in cloud using artificial neural network and adaptive differential evolution". In: *Future Generation Computer Systems* 81 (2018), pp. 41–52. ISSN: 0167-739X. DOI: 10.1016/j.future.2017.10.047.

[LF14] James Lewis and Martin Fowler. *Microservices - a definition of this new architectural term.* https://martinfowler.com/articles/microservices.html. 2014. (Visited on Jan. 1, 2021).

[Li+10] Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. "CloudCmp: comparing public cloud providers". In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement.* 2010, pp. 1–14. DOI: 10.1145/1879141.1879143.

[Lin17] Peter C. Lincoln. "Low Latency Displays for Augmented Reality". PhD thesis. University of North Carolina at Chapel Hill, 2017. DOI: 10.17615/v657-pm26.

[Liu+17] Chunhong Liu, Chuanchang Liu, Yanlei Shang, Shiping Chen, Bo Cheng, and Junliang Chen. "An adaptive prediction approach based on workload pattern discrimination in the cloud". In: *Journal of Network and Computer Applications* 80 (2017), pp. 35–44. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2016.12.017.

[Liu+18] Hang Liu, Fahima Eldarrat, Hanen Alqahtani, Alex Reznik, Xavier De Foy, and Yanyong Zhang. "Mobile Edge Cloud System: Architectures, Challenges, and Approaches". In: *IEEE Systems Journal* 12.3 (Sept. 2018), pp. 2495–2508. ISSN: 1932-8184. DOI: 10.1109/JSYST.2017.2654119.

[LKE17] Chanh Nguyen Le Tan, Cristian Klein, and Erik Elmroth. "Location-aware load prediction in edge data centers". In: *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC).* IEEE. 2017, pp. 25–31. DOI: 10.1109/FMEC.2017.7946403.

[LS16] Yuhua Lin and Haiying Shen. "CloudFog: Leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service". In: *IEEE Transactions on Parallel and Distributed Systems* 28.2 (2016), pp. 431–445. DOI: 10.1109/TPDS.2016.2563428.

[Mah+18] Sumit Maheshwari, Dipankar Raychaudhuri, Ivan Seskar, and Francesco Bronzino. "Scalability and performance evaluation of edge cloud systems for latency constrained applications". In: *2018 IEEE/ACM Symposium on Edge Computing (SEC).* IEEE. 2018, pp. 286–299. DOI: 10.1109/SEC.2018.00028.

[Man+17] Simone Mangiante, Guenter Klas, Amit Navon, Zhuang GuanHua, Ju Ran, and Marco Dias Silva. "VR is on the edge: How to deliver 360 videos in mobile networks". In: *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. 2017, pp. 30–35. ISBN: 9781450350556. DOI: 10.1145/3097895.3097901.

[Meh+16] Amardeep Mehta, William Tärneberg, Cristian Klein, Johan Tordsson, Maria Kihl, and Erik Elmroth. "How beneficial are intermediate layer data centers in mobile edge networks?" In: *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*. IEEE. 2016, pp. 222–229. DOI: 10.1109/FAS-W.2016.55.

[Meh+18] 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.

[MG11] 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.

[Moo25] Henky Ludwell Moore. "A Moving Equilibrium of Demand and Supply". In: *The Quarterly Journal of Economics* 39.3 (May 1925), pp. 357–371. ISSN: 0033-5533. DOI: 10.2307/1882433.

[New15] S. Newman. *Building Microservices*. O'Reilly Media, 2015. ISBN: 9781491950357. URL: https://books.google.se/books?id=1uUDoQEACAAJ.

[Ngu+19] 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 '19. Arlington, Virginia: Association for Computing Machinery, 2019, pp. 250–263. ISBN: 9781450367332. DOI: 10.1145/3318216.3363298.

[NKE] Chanh Nguyen, Cristian Klein, and Erik Elmroth. "State-aware Application Placement in Mobile Edge Clouds". Submitted.

[NKE19] Chanh Nguyen, Cristian Klein, and Erik Elmroth. "Multivariate LSTM-based Location-aware Workload Prediction for Edge Data Centers". In: *CCGrid 2019, 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (IEEE/ACM CCGrid 2019), 14-17 May, Larnaca, Cyprus*. IEEE. 2019, pp. 341–350. DOI: 10.1109/CCGRID.2019.00048.

[NKE20] Chanh Nguyen, Cristian Klein, and Erik Elmroth. “Elasticity Control for Latency-Intolerance Mobile Edge Applications”. In: *Proceedings of the 5th ACM/IEEE Symposium on Edge Computing.* 2020, pp. 70–83. DOI: 10.1109/SEC50012.2020.00013.

[Nun+15] Swaroop Nunna, Apostolos Kousaridas, Mohamed Ibrahim, Markus Dillinger, Christoph Thuemmler, Hubertus Feussner, and Armin Schneider. “Enabling real-time context-aware collaboration through 5G and mobile edge computing”. In: *2015 12th International Conference on Information Technology-New Generations.* IEEE. 2015, pp. 601–605. DOI: 10.1109/ITNG.2015.155.

[OZC18] 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.

[PM17] Jianli Pan and James McElhannon. “Future edge cloud and edge computing for internet of things applications”. In: *IEEE Internet of Things Journal* 5.1 (2017), pp. 439–449. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2767608.

[Por+18] 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.

[Sat+09] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. “The Case for VM-Based Cloudlets in Mobile Computing”. In: *IEEE Pervasive Computing* 8.4 (2009), pp. 14–23. DOI: 10.1109/MPRV.2009.82.

[Sat+13] 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.

[Sat+20] Mahadev Satyanarayanan, Thomas Eiszler, Jan Harkes, Haithem Turki, and Ziqiang Feng. “Edge Computing for Legacy Applications”. In: *IEEE Pervasive Computing* 19.4 (2020), pp. 19–28. ISSN: 1558-2590. DOI: 10.1109/MPRV.2020.3026229.

[SBD18] Meenakshi Syamkumar, Paul Barford, and Ramakrishnan Durairajan. “Deployment Characteristics of "The Edge" in Mobile Edge Computing”. In: *Proceedings of the 2018 Workshop on Mobile Edge Communications.* MECOMM’18. Budapest, Hungary: Association for Computing Machinery, 2018, pp. 43–49. ISBN: 9781450359061. DOI: 10.1145/3229556.3229557. URL: https://doi.org/10.1145/3229556.3229557.

[Shi+16] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. “Edge computing: Vision and challenges”. In: *IEEE internet of things journal* 3.5 (2016), pp. 637–646. ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2579198.

[Ska+18] Per Skarin, William Tärneberg, Karl-Erik Årzen, and Maria Kihl. “Towards mission-critical control at the edge and over 5g”. In: *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE. 2018, pp. 50–57. ISBN: 978-1-5386-7238-9. DOI: 10.1109/EDGE.2018.00014.

[SM20] M Shirer and C MacGillivray. “The Growth in Connected IoT Devices is Expected to Generate 79.4 ZB of Data in 2025, According to a New IDC Forecast”. In: *https://www.idc.com/getdoc.jsp?containerId=prUS45213219* (2020).

[Soa+15] 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.

[Tär+15] William Tärneberg, Amardeep Mehta, Johan Tordsson, Maria Kihl, and Erik Elmroth. “Resource management challenges for the infinite cloud”. In: *10th International Workshop on Feedback Computing at CPSWeek 2015*. 2015.

[Tär+17] 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. DOI: 10.1016/j.future.2016.06.021.

[TLG16] 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.

[Urg+15] Rahul Urgaonkar, Shiqiang Wang, Ting He, Murtaza Zafer, Kevin Chan, and Kin K Leung. “Dynamic service migration and workload scheduling in edge-clouds”. In: *Performance Evaluation* 91 (2015), pp. 205–228. DOI: 10.1016/j.peva.2015.06.013.

[Var+19] 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: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–7. DOI: 10.1109/ICC.2019.8761265.

[VR14] Luis M Vaquero and Luis Rodero-Merino. “Finding your way in the fog: Towards a comprehensive definition of fog computing”. In: *ACM SIGCOMM Computer Communication Review* 44.5 (2014), pp. 27–32. ISSN: 0146-4833. DOI: 10.1145/2677046.2677052.

[Wan+17] Junjue Wang, Brandon Amos, Anupam Das, Padmanabhan Pillai, Norman Sadeh, and Mahadev Satyanarayanan. “A scalable and privacy-aware IoT service for live video analytics”. In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 38–49. ISBN: 9781450350020. DOI: 10.1145/3083187.3083192.

[Wan+18] 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.

[Wan+19] Yuhang Wang, Zhihong Tian, Shen Su, Yanbin Sun, and Chunsheng Zhu. “Preserving location privacy in mobile edge computing”. In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6. DOI: 10.1109/ICC.2019.8761370.

[WZL17] 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.

[YLL18] 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.

[You+10] Andrew J Younge, Gregor Von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, and Warren Carithers. “Efficient resource management for cloud computing environments”. In: *International Conference on Green Computing*. IEEE. 2010, pp. 357–364. ISBN: 978-1-4244-7615-2. DOI: 10.1109/GREENCOMP.2010.5598294.

[You+19] 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.

[Yu16] Yifan Yu. “Mobile edge computing towards 5G: Vision, recent progress, and open challenges”. In: *China Communications* 13.Supplement2 (2016), pp. 89–99. DOI: 10.1109/CC.2016.7833463.

[Zha+18a] Jiale Zhang, Bing Chen, Yanchao Zhao, Xiang Cheng, and Feng Hu. “Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues”. In: *IEEE Access* 6 (2018), pp. 18209–18237. DOI: `10.1109/ACCESS.2018.2820162`.

[Zha+18b] Qingchen Zhang, Laurence T Yang, Zheng Yan, Zhikui Chen, and Peng Li. “An efficient deep learning model to predict cloud workload for industry informatics”. In: *IEEE Transactions on Industrial Informatics* 14.7 (2018), pp. 3170–3178. ISSN: 1941-0050. DOI: `10.1109/TII.2018.2808910`.