

Article

A Survey on Resource Management for Cloud Native Mobile Computing: Opportunities and Challenges

Shih-Yun Huang ¹, Cheng-Yu Chen ² , Jen-Yeu Chen ²  and Han-Chieh Chao ^{2,*}

¹ Department of Computer Science, Tunghai University, Taichung 407224, Taiwan

² Department of Electrical Engineering, National Dong Hwa University, Hualien 97401, Taiwan

* Correspondence: hcchao@gmail.com

Abstract: Fifth-generation mobile communication networks (5G)/Beyond 5G (B5G) can achieve higher data rates, more significant connectivity, and lower latency to provide various mobile computing service categories, of which enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low latency communications (URLLC) are the three extreme cases. A symmetrically balanced mechanism must be considered in advance to fit the different requirements of such a wide variety of service categories and ensure that the limited resource capacity has been properly allocated. Therefore, a new network service architecture with higher flexibility, dispatchability, and symmetrical adaptivity is demanded. The cloud native architecture that enables service providers to build and run scalable applications/services is highly favored in such a setting, while a symmetrical resource allocation is still preserved. The microservice function in the cloud native architecture can further accelerate the development of various services in a 5G/B5G mobile wireless network. In addition, each microservice part can handle a dedicated service, making overall network management easier. There have been many research and development efforts in the recent literature on topics pertinent to cloud native, such as containerized provisioning, network slicing, and automation. However, there are still some problems and challenges ahead to be addressed. Among them, optimizing resource management for the best performance is fundamentally crucial given the challenge that the resource distribution in the cloud native architecture may need more symmetry. Thus, this paper will survey cloud native mobile computing, focusing on resource management issues of network slicing and containerization.



Citation: Huang, S.-Y.; Chen, C.-Y.; Chen, J.-Y.; Chao, H.-C. A Survey on Resource Management for Cloud Native Mobile Computing: Opportunities and Challenges. *Symmetry* **2023**, *15*, 538. <https://doi.org/10.3390/sym15020538>

Academic Editor: José Carlos R. Alcantud

Received: 15 December 2022

Revised: 27 January 2023

Accepted: 15 February 2023

Published: 17 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the deployment and commercial development of fifth-generation mobile communications (5G)/ Beyond 5G (B5G) begin to heat up, many emerging applications, manufacturing, and business models have occurred in the market, such as mixed reality (MR), intelligent manufacturing (IM), and eHealth. These applications benefit from the significant technical requirements provided by 5G, namely, enhanced mobile bandwidth, lower latency, and the ability for massive devices to access wireless networks simultaneously [1–4]. However, although 5G/B5G can provide good quality of service (QoS) to each user equipment (UE), most applications will generate vast and disparate data, making network management difficult. In addition, in the report provided by Cisco [5], we can find that due to the explosive growth of mobile devices in recent years, mobile data traffic will be higher than in the past decade. According to the Next Generation Mobile Networks Alliance (NGMN), operators should meet three requirements: end-to-end system automation and end-to-end system visibility, and system efficiency and management [6].

Furthermore, 5G/B5G is a heterogeneous network (HetNet); that is, it can allow multiple types of networks to exist and access simultaneously, which is very different from

the traditional cellular network. It also leads to management and automation, and the pursuit of efficiency is more complicated than conventional wireless networks. Considering that the traffic and services generated by HetNet are not the same, a good solution is urgently needed to help network operators and service providers to manage mechanisms better; hence, the concept of cloud native was proposed at this time.

In order to better know the cloud native, we must first introduce traditional cloud technology. According to the [7], the concept of cloud architecture is virtualization. Thanks to the virtualization function, the cloud has three advantages for people to use; they are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). We will deploy multiple virtual machines (VMs) in the cloud environment in the physical machine (PM). Each VM can use parallel computing technology to provide computing power, enhancing the overall computing performance of the cloud. Research on the cloud has been a hot topic in this decade, and most of the literature is also focused on the discussion of resource management [8–12], green energy [13–15], and security [16–18]. Although we focus on the debate on cloud native resource management, cloud resource management technologies still have the opportunity to become a solution to the problems encountered by cloud native.

Therefore, we will briefly discuss the state of the art for cloud resource allocation technologies. The concept of symmetry in cloud resource management is also mentioned and explained in detail in [19]. Most of the cloud resource management has achieved symmetry as one of its goals, and load balancing is a method of expressing symmetry. In [20], Jena et al. considered that it should balance all VMs. Hence, they defined the load, energy efficiency, and task priority in detail and used Q-Learning in reinforcement learning to update the p_{best} and g_{best} parameters in the particle swarm algorithm in this research.

Meanwhile, considering the service level agreement (SLA) requirements and task deadlines, Ref. [21] proposed a new load balancing algorithm by QoS and VM priority parameters. On the other hand, the resource management of fog-cloud computing has become a more important topic due to the rise of fog computing in recent years. In [22], considering the latency increased when the IoT device transmitted the data to the cloud. Therefore, if fog computing is used, the latency will greater reduce. However, the resource for fog is limited; hence, resource management happens between fog and cloud. This study also uses the learning method for the management of the resource. In [23], a task scheduling method based on a genetic algorithm (GA) is proposed to solve the resource allocation problem. Finally, Xue et al. focused on the scalability issue in the request scheduling process [24]. In addition to presenting a stochastic preemptive priority queue, they also carefully discussed different cloud environments and architectures.

Ref. [25] presents a taxonomy of resource management techniques and discusses the research challenges these technologies face: energy efficiency, load balancing, hybrid cloud computing, mobile cloud computing, SLA-awareness, network load minimization, and profit maximization. Ref. [26] classified the mobile virtualization techniques and focused on ARM (Advanced RISC Machine) architecture analysis. Virtualization technologies can be categorized as bare-metal and hosted virtualization techniques. In bare-metal virtualization techniques, the hypervisor runs directly over the underlying hardware of the OS; In hosted virtualization techniques, the guest OSs are virtualized over a host OS. They also mention that ARM does not support network function virtualization; it only has general support for I/O virtualization. Therefore, research on network function virtualization customized for mobile devices is a significant future challenge.

Due to the benefits and convenience brought by cloud computing, more and more enterprises are turning their business focus to the application of virtualization technology. Cloud computing uses a lot of virtualization technology to make management easier and save energy. However, the deployment will be less flexible and fast because the prominent architecture uses VMs. Meanwhile, VM will also cost more hardware resources. Containerization technology is widely used in cloud native architecture to reduce the consumption of hardware resources because this technology does not need to simulate hardware resources.

Instead, it focuses on applications to implement microservices more smoothly for fast and flexible deployment. In the 5G environment, it must provide various network services due to the many different network types. Therefore, in the past few years, there have been many studies to analyze the network traffic of various applications and provide optimal resource allocation [27–31], energy efficiency [32–34], and security issues [35–37]. At the same time, many enterprises have gradually realized that the traditional service structure and operation method will usher in significant changes. Enterprises will face the challenge of business transformation because of the diversity of 5G services. Google, AWS, RedHat, VMWare, and other companies formed the cloud native computing foundation (CNCF); the main job is to promote the cloud native architecture so that enterprises can get rid of the dilemma of relying on fixed vendors to provide network services. The cloud native computing foundation (CNCF) is formed by enterprises such as Google, AWS, RedHat, and VMWare. The most straightforward concept in cloud native is designing applications, building microservices, and operating workloads [38]. These workloads are made in the cloud and take advantage of the cloud computing model.

On the other hand, CNCF provides a more detailed explanation of cloud native, which allow organizations to build and execute scalable applications in modern and dynamic environments, including public cloud, private cloud, and hybrid cloud. At the same time, cloud native will also use containerization and microservices. These techniques are loosely bound systems that are resilient, manageable, and observable. Although cloud native will use technologies such as the above, the focus is still on the problems that need to be solved, using the exemplary cloud architecture, developing and running on the cloud to address the limitations of a single architecture. In addition, it will fully utilize the concept of cloud native in mobile communications. The basic idea for cloud native and network architecture of 5G combined with cloud native is shown in Figures 1 and 2. In Figure 2, we know that network virtualization technology and containerization technology can perform network slicing of the 5G core network functions so that each container can run independently and increase the overall network performance [39–41].

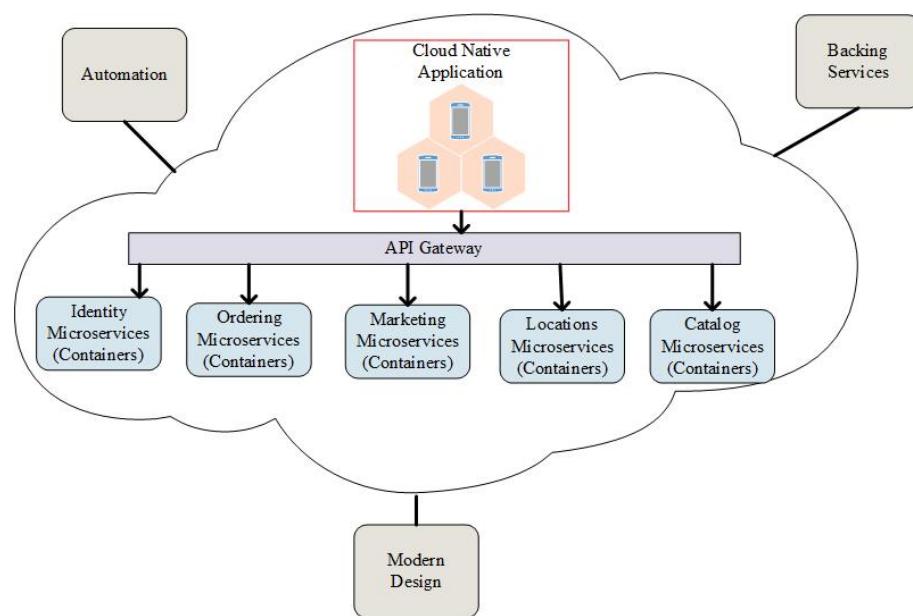


Figure 1. The basic concept for cloud native.

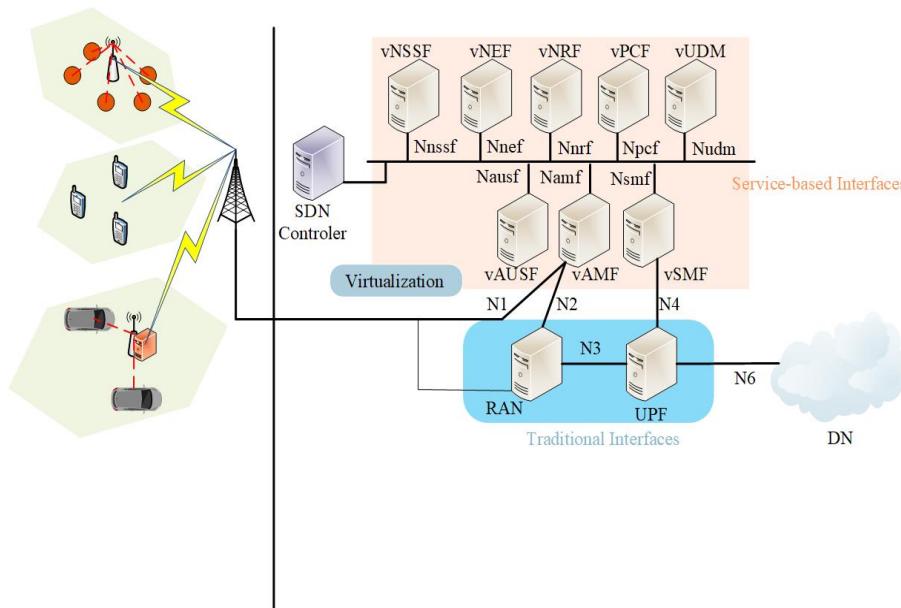


Figure 2. The cloud native architecture with 5G.

Network slicing plays a significant role in next-generation networks. Unlike the traditional one-size-fits-all cellular network architecture, it slices the resource of the physical network infrastructure into dedicated logical networks to provide tailor-made solutions for different application scenarios, and service types [42,43]. Network slicing technology constructs a direct connection path for the cloud and the terminal to optimize service efficiency. In addition, slices are independent; the overload, congestion, and network functions in different slices will not affect each other.

The essential enabling technology of network slicing is network function virtualization (NFV). In 2012, operators initiated the concept of NFV at European Telecommunications Standards Institute (ETSI) [44]; it allows network functions (NFs) to be deployed on commercial servers as software. NFV uses virtualization technology to decouple network function and dedicated hardware to promote the composability and flexibility of network function [45]. However, although the NFV architecture has changed how network functions are realized and deployed, it has mostly stayed the way it was designed [46]. Therefore, using monolithic software virtual network function (VNF) to replace their monolithic hardware-based network functions to implement NFV results in poor use of resources and hinders network agility. For this reason, the concept of cloud native is proposed to avoid these problems [47]. By decomposing a monolithic VNF into a set of cooperating services called “microservices”, the ossification issue with current NFV architectures can be resolved. However, since the distribution of resources in a cloud native architecture may be asymmetrical, the resource management issue of network slicing needs to be discussed in depth.

In addition to containerization, microservices have gradually become one of the critical technologies for cloud native. From [48–50], microservices are an architectural and organizational approach to software development in which software consists of small independent services that communicate through well-defined APIs. In addition, it has the characteristics of spontaneity and specialization. Each component service in the microservice can be freely developed, deployed, operated, and extended without affecting other functions. Furthermore, each service is designed for functional designs that focus on solving a specific problem. To summarize these descriptions, cloud native technology can be deployed quickly and is scalable and resilient. According to the nature of containerization, the distribution of containers over the host platform is asymmetrical and fluctuating because containers can be added, deleted, or migrated over time. Hence, developing an adequate resource allocation scheme has always been critical to cloud native. More and

more literature has conducted in-depth discussions on cloud native technologies, including security, architecture, and resource allocation. According to the cloud native architecture and characteristics, containerization and microservices make development easier. We will focus on cloud native resource allocation issues for sorting and a detailed introduction. Resource allocation has always been an essential topic in wireless networks and clouds. Most of the research will focus on resource management in the cloud, the number of migrations [51,52], the control of VMs, and energy efficiency. This article will be divided into four categories for discussion: resource allocation of containers, resource allocation of microservices, network slicing, and network virtualization technologies.

The hierarchical architecture of the research proposed in this paper is shown in Figure 3, which also contains relevant literature.

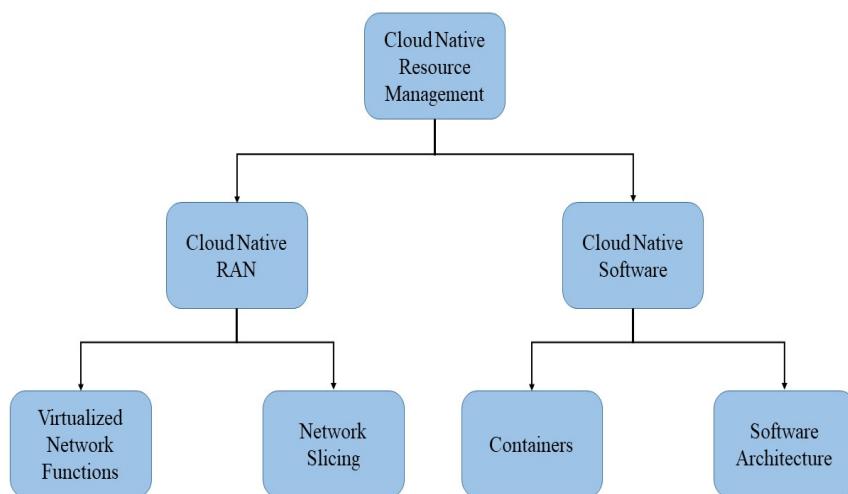


Figure 3. The hierarchical architecture of the research is proposed in this paper. Issues related to virtualized network functions are discussed in [53–61], while issues related to network slicing are covered in [62–71]. In addition, Refs. [72–76] discusses topics related to containers, and [77–81] discusses issues related to resource management with software.

There are four main contributions to this paper:

- (1) We review the latest developments in cloud native technology combined with mobile communication resource allocation.
- (2) We categorize the existing literature from various perspectives, including core networks, service applications, and different technologies.
- (3) We will compare and analyze recent works and discuss their strengths and weaknesses one by one.
- (4) We discuss open issues and challenges in resource allocation for cloud native combined with 5G and unify crucial future research directions.

This paper is organized as follows: Section 2 discusses virtualized network functions and network slicing for resource management in cloud native mobile computing. Section 3 extends resource management to container and software or network architecture. Section 4 provides some critical future research directions for the cloud native subject. We explain the pertinent technologies by carefully selecting the most relative papers in the literature. Finally, Section 5 is the conclusion of this paper.

2. Resource Management of Cloud Native Mobile Computing with RAN

For the 5G cloud native resource management technology, this section will compare and discuss the current related research. This paper will discuss and analyze it in four

parts: network virtualization, network slicing, containerization, and software architecture. First, we see network virtualization of the 5G core network as an important part because network virtualization can significantly reduce the cost of infrastructure and the difficulty of resource management.

2.1. Resource Management for NFV

2.1.1. The Main Challenges with NFV

There are many challenges to implementing NFV, such as portability, performance trade-off, management and orchestration, security, and network stability. Although, in addition, the cloud native network function has been developed due to the advancement of containerization technology, which has improved the system's higher flexibility and scalability, some problems still need to be solved.

2.1.2. The Conventional Solutions NFV

Table 1 shows the comparison of literature on resource management with NFV. In the 5G system, meeting the end QoS is a big challenge. In order to meet the diverse needs of next-generation networks, 5G systems need to be flexible and programmable. On the other hand, meeting end-users' quality of experience (QoE) is another critical challenge. Operators must configure hardware components according to the demand during peak hours to ensure the ultra-low latency and high dynamics of mobile traffic. However, during off-peak hours, idle components result in wasted energy, processing, and network resources. We can solve these problems by adopting NFV. Ref. [53] proposes a QoE-aware elastic execution scheme. The scheme adds the following functions: QoE assessor (QA), Elasticity decision maker (EDM), and resource usage monitor (RUM). These functions are integrated with the service orchestrator and service manager of architecture. The proposed scheme is compatible with existing ETSI NFV architecture and can decide autonomously when and to what extent to implement elasticity. NFV decouples network functions' software components from their respective dedicated hardware. It can optimize the cost of deployment and simplify lifecycle management. Ref. [54] provides a cloud native architecture for mobile cloud networks and is used to implement the designed CN-VNF, a scalable framework for cloud native VNF design.

Configuring cloud native network functions (CNFs) on the edge cloud to build an independent private 5G network can reduce operating costs. However, due to the edge cloud's limited computing power and data storage resources, this distributed processing method will cause CNFs to generate higher backhaul control traffic than legacy NFs. Therefore, to effectively manage CNFs, lightweight control plane management schemes should be designed for a stand-alone private 5G network. Ref. [55] proposes a cloud native network function placement algorithm based on deep Q-network to minimize the cost of backhaul control traffic overhead. Ref. [56] is an extension of the same topic by the same authors and provides a more detailed performance analysis.

Microservices, such as containerized network functions, must process functions with flexible low latency. In addition, to reduce costs, the computing environment should conserve energy. Ref. [57] develops an energy-adaptive network functions framework based on XDP monitoring called X-MAN for managing CPU operational states. In [58], the authors examine cloud native 5G core and its design principles, investigate network slicing and MEC to deliver 5G service-centric use cases, and envision cloud native 5G microservices potential use cases for network slicing. Ref. [59] proposed an intrinsic cloud security (iCS) framework that combines a cloud native environment with a paradigm of moving target defense and mimic defense to achieve secure and reliable network slicing. However, the defensive effectiveness of the system can be severely affected by component heterogeneity, mutation, and recombination strategies. Therefore, a heterogeneous evaluation mechanism needs to be established. Ref. [60] further explores the iCS system that supports heterogeneous resource pool management, which can flexibly set the redundancy rate according to different cost constraints and security levels.

Table 1. Comparison of literature on resource management with NFV.

Ref.	Year	Proposed Method	Tools	Problem
[53]	2016	Autonomic scaling	Ubuntu-14.04.03 LTS	Optimal resource utilization and resource scaling decisions
[54]	2018	5GaaS service architecture and 5G CN-VNF framework	OAI	Review current NFV management solutions
[55]	2020	DQN based algorithm	Not described	Minimize back-haul control traffic cost
[56]	2022	DQN based algorithm	Not described	Minimize back-haul control traffic cost, CNF launching costs, and CNF operating costs
[57]	2022	Monitoring energy-adaptive network functions framework	Ubuntu-20.04 LTS Cisco TRex Docker XDP-Tools Turbostat	Reducing power consumption
[58]	2021	Envisions a cloud native 5G microservices architecture	Kubernetes	Cloud native 5G core study and design
[59]	2021	Intrinsic Cloud Security framework Intrinsic Cloud	DPDK	Security
[60]	2022	Security framework with heterogeneous resource pool management	Not described	Security
[61]	2021	Automate the load balancer deployment	DPDK Cisco TRex	Load balancer deployment

In [61], the author focuses on server load balancing for cloud native architectures and implements a load balancer to manage containerization through Kubernetes easily. This load balancer distributes traffic using eBPF/XDP in the Linux kernel.

2.2. Resource Management for Network Slicing

2.2.1. The Main Challenges with Network Slicing

Network slicing technology meets the diversity and flexibility requirements of 5G networks. It spans the three domains of the transport network, radio access network, and core network and supports customized network services by providing on-demand network slice instances (NSI). However, network slicing technology faces many challenges, like achieving dynamic slice creation and management to maximize benefits, mobility management to meet real-time services, and security issues due to resource sharing between slices. These problems can be summed up as resource management problems encountered in cloud native that combine with network slicing.

2.2.2. The Conventional Solutions Network Slicing

The past mobile networks paradigm will not apply to 5G networks because of the diverse service requirements (eMBB, URLLC, and mMTC). With the technology of NFV, network slicing can use common network infrastructure to meet these different application

and service requirements. Table 2 summarizes and compares the current research on resource management of network slicing.

Table 2. Comparison of literature on resource management with network slicing.

Ref.	Year	Proposed Method	Tools	Problem
[62]	2017	A cloud native approach to network slicing	Linux VMs Django Nginx PostgreSQL OpenStack Apache projects	Lifecycle management
[63]	2021	Validates the MATILDA platform	MATILDA	Lifecycle management
[64]	2021	Uses LSTM RNN model to predict future resource utilization	OpenStack OAI IBN tool	Lifecycle management
[65]	2018	Alternating direction method of multipliers algorithm	Not described	Resource allocation
[66]	2021	Uses LSTM RNN model to predict network load in the future	OAI Kubernetes	Resource allocation for RAN
[67]	2021	Designs a service-oriented RAN architecture	OAI Mosaic5G	Resource allocation for RAN
[68]	2021	DRL algorithm	Kubernetes	Resource allocation and minimize slice migration overhead
[69]	2022	A network slicing management architecture for IIoT applications	Not described	Network slicing management
[70]	2021	Uses components of the slice collection agents in a new framework	OAI Openshift Kubernetes	Network slices monitoring
[71]	2018	Proposes a management and orchestration controller for slice creation	NextEPC CommAgility-SmallCellSTACK-eNodeB	Service guarantees

In [62], the authors introduce cloud native used in the network slicing method that leverages cloud technologies such as NFV, SDN, micro-services, containerization, and cloud native applications. The authors highlight the three-stage lifecycle management of cloud native network slices (design and creation, orchestration and activation, and analytics and optimization) and present cloud native network slicing in a proof-of-concept system. Similarly, for lifecycle management, Ref. [63] gives the platform MATILDA based on cloud native/microservice development principles and introduces an Industry 4.0 application. MATILDA presents a holistic approach to processing the lifecycle of applications' design, orchestration, deployment, and development in a 5G environment. Ref. [64] proposes a network slice lifecycle management solution that can automate the configuration process and perform network slice management and orchestration. The proposed intent-based networking platform uses the long–short time memory (LSTM) RNN model to predict the utilization of future resources.

Ref. [65] proposes a lightweight and flexible network slicing resource allocation framework using cloud native architecture. To ensure the fairness of traffic and computation, the author designed a resource allocation algorithm based on the multiplier alternating direc-

tion method by closely coordinating slice owners, cloud providers, and network controllers, allowing the real-time configuration and automatic scaling of network slices. Monolithic non-configurable hardware devices have dominated RAN in the past few generations of mobile network access. The cloud native approach can even be extended to the RAN through the widespread application of virtualization. Ref. [66] develops and evaluates a machine learning model using an LSTM RNN to predict network load in the future and uses it to decide proactive resource allocation for RAN and core networks. In [67], the author modifies the existing RAN architecture and designs a service-oriented one. It implements scheduling algorithms for multiple slices and the algorithms for user scheduling of the intra-schedulers and inter-schedulers for scheduling micro-SDK.

In the network architecture of network slicing, when encountering mobile events caused by end users, the slice and its allocated resources and services need to be reconfigured for network slicing. Interdependent services and resources must be migrated when slices move between service areas to reduce system overhead and ensure low communication latency for users. However, migrating sliced service instances is a challenging process. Ref. [68] design two algorithms based on deep reinforcement learning (DRL) to select and allocate bandwidth resources and minimize slice migration overhead. Ref. [69] proposes a network-slicing management architecture for IIoT applications such as smart energy, transportation, and factories. In addition, it studies the orchestration architecture of network slice for the IIoT applications for network slicing management and orchestration.

Many resources are included for network slicing in 5G, such as RAN, memory, and computing. Therefore, how to achieve optimal resource management by monitoring resources belonging to these technical domains is an important issue. Ref. [70] introduces a scalable new monitoring framework for 5G network slicing, which employs a novel communication protocol for data collection and supports multi-tenancy in a cloud native environment. Ref. [71] focuses on providing service guarantees, i.e., QoS parameters such as data rate, delay, and slice isolation. To this end, a management and orchestration controller for slice creation is proposed to enable slice tenants to control and manage their respective network slices.

3. Resource Management of Cloud Native Mobile Computing with Software

We discuss cloud native application service resource management in two parts. The first part is the resource management of containerized technology applications. Next is completed by adjusting the cloud native architecture.

3.1. Resource Management for Container

3.1.1. The Main Challenges with Container

Cloud native is a highly feasible and futuristic architecture in 5G mobile communications. In addition to assisting core network virtualization and network slicing, the introduction of containerization technology makes the overall deployment faster, and because each container does not affect the operation of other containers, it is easier to manage and troubleshoot.

At present, the most used container technology is the use of Kubernetes for erection. However, since the container's resources still depend on the resources supported by the physical machine, managing the container resources is still one of the key points to consider. In the following subsections, we will focus on the current research on the resource allocation of containers to discuss and compare their differences one by one.

3.1.2. The Conventional Solutions for Container

Although there have been many studies on cloud native resource management, there are fewer resource management studies on containerization. Most studies still focus on network virtualization and network slicing technologies. Hence, we will organize the current container and microservice management articles to discuss. We summarize and compare the current research with the container for resource management in Table 3.

Considering the cloud native performance in [72], the authors propose using a monitoring system to collect real-time resource usage for managing containers or microservices. At the same time, they also calculate the completion time of containerized big data and deep learning applications on Docker and Kubernetes platforms, respectively. Because multiple workers are usually used as computing nodes in a native cloud environment, the container placement strategies will have different performance results. In Docker Swarm, a new container is added to the node with the fewest running containers, a container is placed on the entire node in the cluster, or a worker is randomly selected to build a container. In Kubernetes, a scoring algorithm is usually used, and the most suitable node is determined according to various factors, such as available resources. By using this method, it can reduce the time to completion by changing the default configuration. Since cloud native is a flexible and dynamic system, continuously monitoring the microservice system is a significant challenge.

Table 3. Comparison of literature on resource management with containers.

Ref.	Year	Proposed Method	Tools	Problem
[72]	2020	Monitoring system and analyze completion times	Docker and Kubernetes	Performance analysis and container strategies
[73]	2018	Quantized MPI	Docker and Singularity	Performance analysis
[74]	2019	Self-adaptive resource sharing	Kubernetes	Vertical container expansion
[75]	2018	Use iperf3 for analysis	AWS	Containers and performance
[76]	2018	CloudRanger	IBM Bluemix	Error detection

The number of microservices will increase or decrease depending on the situation, and the workload will also be affected [76]. In addition, containerization technology will also seriously affect the overall performance. Cloud native can stop, restart, or move a container from an existing node to another node. This action will lead to weak contextual correlation, making it difficult to track the state of the container. Therefore, to meet this challenge, they propose a new system architecture: CloudRanger. This method uses the dynamic causal relationship analysis to construct the influence diagram between applications. Meanwhile, considering the situation of diagnosing the occurrence of events in the cloud native system, they also use the second-order random walk-based heuristic investigation algorithm to identify the service in question.

In order to facilitate the management and control of container resources and performance, the performance analysis of Docker and Singularity on Chameleon bare metal nodes is introduced in [73]. The performance indicators are CPU, memory, and delay sensitive, respectively. Docker can communicate with InfiniBand through RDMA communication, the mechanism by which the hardware does the mapping. Next, they also present the analysis of parallel workload mapping elements. Through this research, we can judge how to choose the appropriate workload container technology and method when needed. At the same time, they also use Docker to orchestrate containers, including a container for each host to have the same IP but different ports. Then, the second orchestration method is to use Docker Swarm to create an overlay network that spans multiple nodes so that all containers under the same subnet address can be assigned a separate IP. The third method uses numerous containers for each node to connect to the public overlay network. Since microservices will affect workload changes, operators providing cloud services can use vertical container scaling, adding or removing data.

To avoid violating service level objectives (SLOs) and increasing the utilization of extra resources, Podolskiy et al. [74] first let the system learn the correlation model between the SLO and workload, service level indicator, and resource limitation. Then, to meet the limit of the SLO, they obtain the most suitable solution through optimization and the

brute-force attack. Finally, to reduce its resource consumption, they use this solution to scale the container to achieve effective resource management vertically.

Since containerization technology is continuously used in cloud native, we can deploy the overall system more flexibly. Furthermore, according to [75], they point out that a container network between hosts is composed of any containerized software system on multiple hosts. Therefore, although much research has conducted in-depth discussion and evaluation of the performance of the local network, there needs to be a network performance evaluation between the container networks. Hence, they use iperf3 to evaluate the network performance and system of the public cloud for different container networks. Ultimately, they find that the overall performance difference would be significant when transferring data volumes greater than 5Gbps and requiring network encryption. We can certainly use this conclusion for resource management strategies.

3.2. Resource Management for Software Architecture

3.2.1. The Main Challenges with Software Architecture

In a cloud native environment, it will achieve flexibility and highly reliable scheduling by shutting down, adding or moving containers, and using microservices; this will cause management difficulties. On the other hand, cloud native adaptive adjustment methods and architectures have always been the research focus. Using adaptive architectures can effectively improve system performance and resource management.

3.2.2. The Conventional Solutions for Software Architecture

In this subsection, we will introduce research on resource management by utilizing different architectures, and Table 4 compares literature on resource management with software architecture. In [77], considering that the 5G cloud native architecture can reduce its deployment cost by virtualizing the core network, the authors propose a scalable cloud native architecture called a cloud native solution for a mobility management entity. This architecture is mainly a data production center based on a microservice architecture. The advantages of this architecture are high scalability and support for automatically scaling up and down the required microservices, enabling the overall system to achieve load balancing. They first extend the NFV-LTE-EPC framework and then use the open-source orchestrator Kubernetes and the Docker container platform to complete the goal of network function virtualization. Then, use the monitoring tool Prometheus to obtain network information. In order to achieve load balancing, they also design an L7 load balancer for this architecture. Using this, it can store the current state of MME in centralized data. It is better than the traditional L4 load balancer regarding throughput, load balancing, adaptability, and computing resources.

Table 4. Comparison of literature on resource management with software architecture.

Ref.	Year	Proposed Method	Tools	Problem
[77]	2017	CNS-MME	Docker and Kubernetes	Load balance
[78]	2020	AMoCNA	VM	Self-management
[79]	2019	Kubow	Kubernetes	Self adaptive
[80]	2020	MicroRCA	Kubernetes	Performance diagnosis for microservices
[81]	2020	Use MLP to predict load	Not described	Workloads for ML

On the other hand, Ref. [78] also defines the self-management requirements of cloud native applications in detail, and they also realize the automatic function through advanced strategies in the research. Its primary method is implementing the application self-management framework (AMoCNA) through model-driven architecture. At the same time, this architecture is divided into five layers for separate discussions: instrumentation

layer, observation layer, management layer, inference layer, and control layer. When using this architecture, the cloud native management's complexity can be effectively reduced, making resource management more efficient and straightforward. Since Kubernetes is the most commonly used open-source software for cloud native architecture, in [79], the authors propose an adaptive service system based on this software called Kubow. Its primary implementation method is to complete through customization and the Rainbow adaptive framework so that this system can run on Docker containers and Kubernetes. To integrate Kubow and Kubernetes, they first define the architecture through Acme based on containerized services and connector collections. Then, the authors describe it through two component types, DeploymentT and ServiceT, and two connector types, LabelSelectorConnectorT and ServiceConnectorT. It can perform service and deployment through two component types, and we can understand the relationship established between different services and resources in Kubernetes by using two connector types. In addition, they also mention that according to different application model architectures, the definition of Kubow will change, that is, monitoring, policy, change, etc.

In [80], microservices are discussed in detail. However, because it will frequently update microservices and infrastructure in cloud native, it will encounter considerable challenges in meeting the problem diagnosis in self-adaptation. To solve this problem, they propose MicroRCA technology. This method is based on the performance symptoms of the program and the corresponding resource utilization rate to judge; that is to say, using this method does not require program detection. At the same time, because this method will construct an exception propagation graph across services and machines, it can be well adapted to different types of microservices and simplify overall resource management.

Machine learning, deep learning, and reinforcement algorithms have been increasing in recent years. Machine learning allows researchers or engineers to quickly make predictions and approximate optimal answers to research-related questions. However, considering that the current machine learning uses the mighty computing power of the cloud for learning, at the same time, due to changes in the software service architecture, machine learning is gradually moving to run on cloud native. The flexible deployment and scalability of cloud native solve the resources that machine learning requires. However, although cloud native features and advantages can assist machine learning operations, they still encounter load-balancing problems. In [81], to solve this problem, they first use the AI4DL framework to define the workload, observe the resource consumption status, and then use the temporal multi-layer perceptron to make predictions for different types of workloads in the container. At the same time, they also propose a predictive vertical autoscaling strategy to resize containers dynamically. Considering the high dynamics of cloud native and microservices, if the container is adjusted, even if the predicted result is a small change, it will cause unnecessary operations and management difficulties. Therefore, the container adjustment will only occur when a significant change is predicted.

4. The Challenge and Future Trends

The software architecture is gradually changing from a single to a microservice architecture. Many containerization technologies are applied to cloud native, and our lives are progressively evolving. On the other hand, it will also transform the traditional core network into a virtual core network through the maturity of technologies, such as containerization technology, microservices, network slicing, and virtualized network functions. In the evolution of these technologies, cloud native has played a significant role. Although we compared and discussed resource management for virtualized network functions, network slicing, container and microservice management, and adaptive architecture in cloud native edge computing, these are only part of cloud native. Cloud native still has several significant challenges to overcome. First, because cloud native provides flexibility and scalability for deployment, the overall management and control will be more complicated than traditional cloud.

In the view of microservice, it will be deleted or added over time, seriously affecting the system performance and difficulty in detecting faults. At the same time, due to the use of containerization technology, the opening/closing of containers and moving to other nodes all require a comprehensive strategy and management. In addition, we still need to consider issues, such as limited resource control, scheduling, and automatic scaling for containerization technology. The tools or methods with the ability to monitor data are widely used in most studies to solve the problem of the cloud native platform [82,83].

Virtualization technology with virtual machines (VMs) is the foundation of cloud computing. It provides flexible and resilient information technology (IT) infrastructure for cloud services. However, using VMs may have high energy consumption and waste computing resources due to running the same operations by various guest operating systems [84]. Given this, they introduced container technology to solve the energy consumption and computing resource-wasting problem. In addition, container technology improves resource usage efficiency by sharing the same infrastructure with the same operating system. Cloud native environments can leverage hardware acceleration to offload 5G RAN functions. However, the software implementation of low-density parity-check (LDPC) decoding in the 5G physical layer is challenging due to its iterative and complex processing. Therefore, it may need to consume very high power to achieve the expected performance of 5G mobile networks. Ref. [85] proposed a dynamically activating LDPC decoding on Kubernetes field-programmable gate array (FPGA) method to accelerate the 5G RAN distributed unit (DU) stack of cloud native.

Next, the cloud native platform is also vulnerable to attacks due to its characteristics; hence, security issues still need attention. In [86], the SmartX Multi-tire security is proposed. This approach will leverage monitoring, visualization, and filtering network topology traffic at all levels for robust network security. Considering the gradual expansion of security vulnerabilities due to containerization technology, static security mechanisms are insufficient to ensure the security of data or systems; therefore, internal security mechanisms must also be considered. Meanwhile, the IoT will be the primary trend of most network applications in the future, generating many new microservices. These microservices need to be distinguished according to different application scenarios, for example, how to organize themselves or serve to explore. At the same time, whether to utilize the cloud or edge computing will also have to be considered.

5. Conclusions

With the development of mobile communication and smart devices, 5G communication has gradually become a new focus. According to the 5G mobile communication architecture, it will be a heterogeneous network with high complexity and complex management; it can access different network types simultaneously. For this reason, operators need a cloud environment that they can dynamically adjust or a new kind of cloud architecture closer to the end of the network. Fortunately, the software architecture has gradually moved closer to microservices and containerization in recent years. The core network used by 5G also introduces these technologies to virtualize core network functions and implement network slicing on the radio access network, gradually realizing the cloud native architecture. This paper conducts a detailed analysis and comparison of resource management for cloud native edge computing. We discussed resource management with different technologies: virtualized network functions, network slicing, containerization technology, and software architecture. We also proposed to readers the future research trends of cloud native and other challenges in this article, hoping that more researchers will pay attention to cloud native issues.

Author Contributions: Co-first Author, C.-Y.C.; Conceptualization, H.-C.C., J.-Y.C., C.-Y.C. and S.-Y.H.; methodology, S.-Y.H., C.-Y.C.; software, J.-Y.C.; validation, J.-Y.C.; investigation, J.-Y.C., C.-Y.C. and S.-Y.H.; data curation, J.-Y.C.; writing—original draft preparation, S.-Y.H. and C.-Y.C.; writing—review and editing, H.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This research was partly funded by the Ministry of Science and Technology of the R.O.C. under grants NSTC 111-2221-E-259-007.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Xu, Y.; Gui, G.; Gacanin, H.; Adachi, F. A survey on resource allocation for 5G heterogeneous networks: Current research, future trends, and challenges. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 668–695. [[CrossRef](#)]
- Tang, Y.; Dananjayan, S.; Hou, C.; Guo, Q.; Luo, S.; He, Y. A survey on the 5G network and its impact on agriculture: Challenges and opportunities. *Comput. Electron. Agric.* **2021**, *180*, 105895. [[CrossRef](#)]
- Dangi, R.; Lalwani, P.; Choudhary, G.; You, I.; Pau, G. Study and investigation on 5G technology: A systematic review. *Sensors* **2021**, *22*, 26. [[CrossRef](#)]
- Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Ylianttila, M. A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 1160–1192. [[CrossRef](#)]
- Cisco, U. *Cisco Annual Internet Report (2018–2023) white Paper*; Cisco: San Jose, CA, USA, 2020.
- Yastrebova, A.; Kirichek, R.; Koucheryavy, Y.; Borodin, A. Koucheryavy. Future Networks 2030: Architecture & Requirements. In Proceedings of the 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 5–9 November 2018; pp. 1–8.
- Yoo, C.S. Cloud computing: Architectural and policy implications. *Rev. Ind. Organ.* **2011**, *38*, 405–421. [[CrossRef](#)]
- Parikh, S.M. A survey on cloud computing resource allocation techniques. In Proceedings of the 2013 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, Gujarat, India, 28–30 November 2013; pp. 1–5.
- Kumar, P.; Kumar, R. Issues and challenges of load balancing techniques in cloud computing: A survey. *AcM Comput. Surv. (CSUR)* **2019**, *51*, 1–35. [[CrossRef](#)]
- Afzal, S.; Kavitha, G. Load balancing in cloud computing—A hierarchical taxonomical classification. *J. Cloud Comput.* **2019**, *8*, 22. [[CrossRef](#)]
- Gill, S.S.; Garraghan, P.; Stankovski, V.; Casale, G.; Thulasiram, R.K.; Ghosh, S.K.; Ramamohanarao, K.; Buyya, R. Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge. *J. Syst. Softw.* **2019**, *155*, 104–129. [[CrossRef](#)]
- Madni, S.H.H.; Latiff, M.S.A.; Coulibaly, Y.; Abdulhamid, S.M. Recent advancements in resource allocation techniques for cloud computing environment: A systematic review. *Clust. Comput.* **2017**, *20*, 2489–2533. [[CrossRef](#)]
- Xu, M.; Toosi, A.N.; Buyya, R. A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing. *IEEE Trans. Sustain. Comput.* **2020**, *6*, 544–558. [[CrossRef](#)]
- Tuli, S.; Ilager, S.; Ramamohanarao, K.; Buyya, R. Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. *IEEE Trans. Mob. Comput.* **2020**, *21*, 940–954. [[CrossRef](#)]
- Marahatta, A.; Pirbhulal, S.; Zhang, F.; Parizi, R.M.; Choo, K.K.R.; Liu, Z. Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center. *IEEE Trans. Cloud Comput.* **2019**, *9*, 1376–1390. [[CrossRef](#)]
- Awaysheh, F.M.; Aladwan, M.N.; Alazab, M.; Alawadi, S.; Cabaleiro, J.C.; Pena, T.F. Security by design for big data frameworks over cloud computing. *IEEE Trans. Eng. Manag.* **2021**, *69*, 3676–3693. [[CrossRef](#)]
- Alouffi, B.; Hasnain, M.; Alharbi, A.; Alosaimi, W.; Alyami, H.; Ayaz, M. A systematic literature review on cloud computing security: Threats and mitigation strategies. *IEEE Access* **2021**, *9*, 57792–57807. [[CrossRef](#)]
- Nhlabatsi, A.; Hong, J.B.; Kim, D.S.; Fernandez, R.; Hussein, A.; Fetais, N.; Khan, K.M. Threat-specific security risk evaluation in the cloud. *IEEE Trans. Cloud Comput.* **2018**, *9*, 793–806. [[CrossRef](#)]
- Varshney, P.; Simmhan, Y. Characterizing application scheduling on edge, fog, and cloud computing resources. *Softw. Pract. Exp.* **2020**, *50*, 558–595. [[CrossRef](#)]
- Jena, U.; Das, P.; Kabat, M. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. *J. King Saud-Univ.-Comput. Inf. Sci.* **2020**, *34*, 2332–2342. [[CrossRef](#)]
- Shafiq, D.A.; Jhanji, N.Z.; Abdullah, A.; Alzain, M.A. A load balancing algorithm for the data centres to optimize cloud computing applications. *IEEE Access* **2021**, *9*, 41731–41744. [[CrossRef](#)]
- Abbasi, M.; Yaghoobikia, M.; Rafiee, M.; Jolfaei, A.; Khosravi, M.R. Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems. *Comput. Commun.* **2020**, *153*, 217–228. [[CrossRef](#)]
- Duan, K.; Fong, S.; Siu, S.W.; Song, W.; Guan, S.S.U. Adaptive incremental genetic algorithm for task scheduling in cloud environments. *Symmetry* **2018**, *10*, 168. [[CrossRef](#)]

24. Xue, C.; Lin, C.; Hu, J. Scalability analysis of request scheduling in cloud computing. *Tsinghua Sci. Technol.* **2019**, *24*, 249–261. [[CrossRef](#)]
25. Mustafa, S.; Nazir, B.; Hayat, A.; Madani, S.A. Resource management in cloud computing: Taxonomy, prospects, and challenges. *Comput. Electr. Eng.* **2015**, *47*, 186–203. [[CrossRef](#)]
26. Shuja, J.; Gani, A.; Bilal, K.; Khan, A.U.R.; Madani, S.A.; Khan, S.U.; Zomaya, A.Y. A survey of mobile device virtualization: Taxonomy and state of the art. *Acm Comput. Surv. (CSUR)* **2016**, *49*, 1–36. [[CrossRef](#)]
27. Peng, M.; Wang, C.; Li, J.; Xiang, H.; Lau, V. Recent advances in underlay heterogeneous networks: Interference control, resource allocation, and self-organization. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 700–729. [[CrossRef](#)]
28. Gatti, R.; Shankar, S.; Murthy, K. Effects of bidirectional resource allocation schemes for advanced long-term evolution system in heterogeneous networks. *Int. J. Commun. Netw. Distrib. Syst.* **2021**, *27*, 241–258. [[CrossRef](#)]
29. Zhang, J.; Xia, W.; Yan, F.; Shen, L. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing. *IEEE Access* **2018**, *6*, 19324–19337. [[CrossRef](#)]
30. Khalili, A.; Akhlaghi, S.; Tabassum, H.; Ng, D.W.K. Joint user association and resource allocation in the uplink of heterogeneous networks. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 804–808. [[CrossRef](#)]
31. Cho, H.H.; Lai, C.F.; Shih, T.K.; Chao, H.C. Learning-based Data Envelopment Analysis for External Cloud Resource Allocation. *ACM/Springer Mob. Netw. Appl. (MONET)* **2016**, *21*, 846–855. [[CrossRef](#)]
32. Khan, W.U.; Li, X.; Ihsan, A.; Ali, Z.; Elhalawany, B.M.; Sidhu, G.A.S. Energy efficiency maximization for beyond 5G NOMA-enabled heterogeneous networks. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3250–3264. [[CrossRef](#)]
33. Shuvo, M.S.A.; Munna, M.A.R.; Sarker, S.; Adhikary, T.; Razzaque, M.A.; Hassan, M.M.; Aloj, G.; Fortino, G. Energy-efficient scheduling of small cells in 5G: A meta-heuristic approach. *J. Netw. Comput. Appl.* **2021**, *178*, 102986. [[CrossRef](#)]
34. Giannopoulos, A.; Spantideas, S.; Kapsalis, N.; Karkazis, P.; Trakadas, P. Deep reinforcement learning for energy-efficient multi-channel transmissions in 5G cognitive hetnets: Centralized, decentralized and transfer learning based solutions. *IEEE Access* **2021**, *9*, 129358–129374. [[CrossRef](#)]
35. Park, J.H.; Rathore, S.; Singh, S.K.; Salim, M.M.; Azzaoui, A.; Kim, T.W.; Pan, Y.; Park, J.H. A comprehensive survey on core technologies and services for 5G security: Taxonomies, issues, and solutions. *Hum.-Centric Comput. Inf. Sci.* **2021**, *11*, 36.
36. Lal, N.; Tiwari, S.M.; Khare, D.; Saxena, M. Prospects for handling 5G network security: Challenges, recommendations and future directions. *J. Phys. Conf. Ser.* **2021**, *1714*, 012052. [[CrossRef](#)]
37. Sullivan, S.; Brightente, A.; Kumar, S.; Conti, M. 5G security challenges and solutions: A review by OSI layers. *IEEE Access* **2021**, *9*, 116294–116314. [[CrossRef](#)]
38. Gannon, D.; Barga, R.; Sundaresan, N. Cloud-native applications. *IEEE Cloud Comput.* **2017**, *4*, 16–21. [[CrossRef](#)]
39. Arouk, O.; Nikaein, N. 5g cloud-native: Network management & automation. In Proceedings of the NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–2.
40. Ziegler, V.; Viswanathan, H.; Flinck, H.; Hoffmann, M.; Räisänen, V.; Hätkönen, K. 6G architecture to connect the worlds. *IEEE Access* **2020**, *8*, 173508–173520. [[CrossRef](#)]
41. Kukliński, S.; Tomaszewski, L.; Kołakowski, R.; Chemouil, P. 6G-LEGO: A framework for 6G network slices. *J. Commun. Netw.* **2021**, *23*, 442–453. [[CrossRef](#)]
42. Zhang, S. An overview of network slicing for 5G. *IEEE Wirel. Commun.* **2019**, *26*, 111–117. [[CrossRef](#)]
43. Nokia. *Dynamic End-to-End Network Slicing for 5G*; White Paper: Espoo, Finland, 2016.
44. ETSI, G. Network functions virtualisation (nfv): Architectural framework. *ETSI Gs Nfv* **2013**, *2*, V1.
45. Zhang, Y. *Network Function Virtualization: Concepts and Applicability in 5G Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
46. Duan, Q. Intelligent and autonomous management in cloud-native future networks—A survey on related standards from an architectural perspective. *Future Internet* **2021**, *13*, 42. [[CrossRef](#)]
47. Brown, G. Designing Cloud-Native 5G Core Networks. February 2017. Heavy Reading. Available online: <https://www.scribd.com/document/358153029/Nokia-5g-Core-White-Paper> (accessed on 14 February 2023).
48. Thönes, J. Microservices. *IEEE Softw.* **2015**, *32*, 116. [[CrossRef](#)]
49. Balalaie, A.; Heydarnoori, A.; Jamshidi, P. Microservices architecture enables devops: Migration to a cloud-native architecture. *IEEE Softw.* **2016**, *33*, 42–52. [[CrossRef](#)]
50. Jamshidi, P.; Pahl, C.; Mendonça, N.C.; Lewis, J.; Tilkov, S. Microservices: The journey so far and challenges ahead. *IEEE Softw.* **2018**, *35*, 24–35. [[CrossRef](#)]
51. Linthicum, D.S. Cloud-native applications and cloud migration: The good, the bad, and the points between. *IEEE Cloud Comput.* **2017**, *4*, 12–14. [[CrossRef](#)]
52. Osmani, L.; Kauppinen, T.; Komu, M.; Tarkoma, S. Multi-cloud connectivity for kubernetes in 5g networks. *IEEE Commun. Mag.* **2021**, *59*, 42–47. [[CrossRef](#)]
53. Dutta, S.; Taleb, T.; Ksentini, A. QoE-aware elasticity support in cloud-native 5G systems. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
54. Imadali, S.; Bousselmi, A. Cloud native 5g virtual network functions: Design principles and use cases. In Proceedings of the 2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2), Paris, France, 19–22 November 2018; pp. 91–96.

55. Kim, J.; Lee, J.; Kim, T.; Pack, S. Deep reinforcement learning based cloud-native network function placement in private 5g networks. In Proceedings of the 2020 IEEE Globecom Workshops (GC Wkshps. IEEE), Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
56. Kim, J.; Lee, J.; Kim, T.; Pack, S. Deep Q-Network-based Cloud-Native Network Function Placement in Edge Cloud-Enabled Non-Public Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**. [[CrossRef](#)]
57. Xiang, Z.; Höweler, M.; You, D.; Reisslein, M.; Fitzek, F.H. X-MAN: A non-intrusive power manager for energy-adaptive cloud-native network functions. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 1017–1035. [[CrossRef](#)]
58. Shah, S.D.A.; Gregory, M.A.; Li, S. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. *IEEE Access* **2021**, *9*, 10903–10924. [[CrossRef](#)]
59. Qiang, W.; Chunming, W.; Xincheng, Y.; Qiumei, C. Intrinsic security and self-adaptive cooperative protection enabling cloud native network slicing. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1287–1304. [[CrossRef](#)]
60. Wu, Q.; Wang, R.; Yan, X.; Wu, C.; Lu, R. Intrinsic Security: A Robust Framework for Cloud-Native Network Slicing via a Proactive Defense Paradigm. *IEEE Wirel. Commun.* **2022**, *29*, 146–153. [[CrossRef](#)]
61. Lee, J.B.; Yoo, T.H.; Lee, E.H.; Hwang, B.H.; Ahn, S.W.; Cho, C.H. High-performance software load balancer for cloud-native architecture. *IEEE Access* **2021**, *9*, 123704–123716. [[CrossRef](#)]
62. Sharma, S.; Miller, R.; Francini, A. A cloud-native approach to 5G network slicing. *IEEE Commun. Mag.* **2017**, *55*, 120–127. [[CrossRef](#)]
63. Bolla, R.; Bruschi, R.; Burow, K.; Davoli, F.; Ghairi, Z.; Gouvas, P.; Lombardo, C.; Pajo, J.F.; Zafeiropoulos, A. From cloud-native to 5g-ready vertical applications: An industry 4.0 use case. In Proceedings of the 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), Paris, France, 7–10 June 2021; pp. 1–6.
64. Abbas, K.; Khan, T.A.; Afaq, M.; Song, W.C. Network slice lifecycle management for 5g mobile networks: An intent-based networking approach. *IEEE Access* **2021**, *9*, 80128–80146. [[CrossRef](#)]
65. Leconte, M.; Paschos, G.S.; Mertikopoulos, P.; Kozat, U.C. A resource allocation framework for network slicing. In Proceedings of the IEEE INFOCOM 2018–IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 2177–2185.
66. Mudvari, A.; Makris, N.; Tassiulas, L. ML-driven scaling of 5G Cloud-Native RANs. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–6.
67. Schmidt, R.; Nikaein, N. RAN engine: Service-oriented RAN through containerized micro-services. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 469–481. [[CrossRef](#)]
68. Boudi, A.; Bagaa, M.; Pöyhönen, P.; Taleb, T.; Flinck, H. AI-based resource management in beyond 5G cloud native environment. *IEEE Netw.* **2021**, *35*, 128–135. [[CrossRef](#)]
69. Wu, Y.; Dai, H.N.; Wang, H.; Xiong, Z.; Guo, S. A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory. *IEEE Commun. Surv. Tutorials* **2022**, *24*, 1175–1211. [[CrossRef](#)]
70. Mekki, M.; Arora, S.; Ksentini, A. A Scalable Monitoring Framework for Network Slicing in 5G and Beyond Mobile Networks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 413–423. [[CrossRef](#)]
71. Bektas, C.; Monhof, S.; Kurtz, F.; Wietfeld, C. Towards 5G: An empirical evaluation of software-defined end-to-end network slicing. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
72. Mao, Y.; Fu, Y.; Gu, S.; Vhaduri, S.; Cheng, L.; Liu, Q. Resource management schemes for cloud-native platforms with computing containers of docker and kubernetes. *arXiv* **2020**, arXiv:2010.10350.
73. Saha, P.; Beltre, A.; Uminski, P.; Govindaraju, M. Evaluation of docker containers for scientific workloads in the cloud. In Proceedings of the Practice and Experience on Advanced Research Computing, Pittsburgh, PA, USA, 22–26 July 2018; pp. 1–8.
74. Podolskiy, V.; Mayo, M.; Koay, A.; Gerndt, M.; Patros, P. Maintaining SLOs of cloud-native applications via self-adaptive resource sharing. In Proceedings of the 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Umea, Sweden, 16–20 June 2019; pp. 72–81.
75. Bankston, R.; Guo, J. Performance of container network technologies in cloud environments. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; pp. 277–283.
76. Wang, P.; Xu, J.; Ma, M.; Lin, W.; Pan, D.; Wang, Y.; Chen, P. Cloudranger: Root cause identification for cloud native systems. In Proceedings of the 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Washington, DC, USA, 1–4 May 2018; pp. 492–502.
77. Amogh, P.; Veeramachaneni, G.; Rangisetti, A.K.; Tamma, B.R.; Franklin, A.A. A cloud native solution for dynamic auto scaling of MME in LTE. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–7.
78. Kosińska, J.; Zieliński, K. Autonomic management framework for cloud-native applications. *J. Grid Comput.* **2020**, *18*, 779–796. [[CrossRef](#)]
79. Aderaldo, C.M.; Mendonça, N.C.; Schmerl, B.; Garlan, D. Kubow: An architecture-based self-adaptation service for cloud native applications. In Proceedings of the 13th European Conference on Software Architecture, Paris, France, 9–13 September 2019; Volume 2, pp. 42–45.

80. Wu, L.; Tordsson, J.; Elmroth, E.; Kao, O. Microrca: Root cause localization of performance issues in microservices. In Proceedings of the NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–9.
81. Buchaca, D.; Berral, J.L.; Wang, C.; Youssef, A. Proactive container auto-scaling for cloud native machine learning services. In Proceedings of the 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), Virtual Event, 18–24 October 2020; pp. 475–479.
82. Henning, S.; Hasselbring, W. A configurable method for benchmarking scalability of cloud-native applications. *Empir. Softw. Eng.* **2022**, *27*, 1–42. [[CrossRef](#)]
83. Barrachina-Muñoz, S.; Payaró, M.; Mangues-Bafalluy, J. Cloud-native 5G experimental platform with over-the-air transmissions and end-to-end monitoring. In Proceedings of the 2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2022; pp. 692–697.
84. Jayalakshmi, S.; Bharanidharan, G.; Jayalakshmi, S. Energy Efficient Next-Gen of Virtualization for Cloud-native Applications in Modern Data Centres. In Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 7–9 October 2020; pp. 203–210.
85. Dion, J.; Lallet, J.; Beaulieu, L.; Savelli, P.; Bertin, P. Cloud Native Hardware Accelerated 5G virtualized Radio Access Network. In Proceedings of the 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 13–16 September 2021; pp. 1061–1066.
86. Shin, J.S.; Kim, J. SmartX Multi-Sec: A Visibility-Centric Multi-Tiered Security Framework for Multi-Site Cloud-Native Edge Clusters. *IEEE Access* **2021**, *9*, 134208–134222. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.