Mobile Cloudization Storytelling: Taxonomy and Current Issues from Optimization Perspective

Nhu-Ngoc Dao, Woongsoo Na, Sungrae Cho

***Abstract*—Next generation mobile cloud is expected to support billions of connected things. To deal with this requirement, clouds have evolved to utilize multitiered cloudization infrastructure to provide computing, caching, and networking resources throughout entire networks. Cloudization consists of cloud, fog, edge, and peer-to-peer (P2P) computing capabilities at the core, distribution, access, and peer-aware networks, respectively. For distinct objectives, utilization of the cloudization infrastructure is optimized accordingly. This article considers the cloudization framework for next-generation mobile computing from an optimization perspective. First, a comprehensive cloudization architecture is analyzed on the basis of European telecommunications standards institute (ETSI) network functions virtualization management and orchestration (NFV-MANO) specifications. Second, current effective approaches are classified, and the optimization objectives are discussed. Third, intrinsic computing on user devices is described as an extension of cloudization. Last, recent issues are clarified for future research directions.**

***Index Terms*—mobile cloudization, mobile computing, offloading optimization**

I. Introduction

The recently emerging IoTization paradigm provides billions of things with Internet ability in next-generation mobile networks. These connected things (mostly including smartphones, machine-to-machine communication devices, and wearable devices) are predicted to exponentially increase mobile data traffic up to 49 exabytes per month within the next three years [1]. As a result, mobile data traffic is characterized by heterogeneity, mobility, and massiveness that in turn force network operators to pay a considerable amount of attention to next-generation mobile computing services.

To overcome these surging requirements, there is an expansion of cloud computing capabilities from the core to the edge during mobile network evolution [2], referred to as *cloudization*. As the core enabler for effective mobile computing, cloudization provides computing, caching, and networking infrastructures at the core, distribution, access, and peer-aware networks, namely cloud, fog, edge, and peer-to-peer (P2P) computing, respectively (see Fig. 1). Despite introducing the same services, these four computing tiers offer distinguished features in terms of capability and performance. Each computing tier plays a unique role, and they are complementary to fulfill multitiered cloudization infrastructure.

In particular, since each user device or application prioritizes its own requirements in terms of the latency, energy, availability, resource consumption, and so on, cloudization utilization must be optimized for user demands [3]. This leads to the need for flexible cloudization orchestration among computing tiers. In this circumstance, network slicing techniques [4] are adopted as an effective solution to virtually separate computing resources for each optimal objective. For instance, latency minimization for time-sensitive services should be mainly allocated with edge and fog computing resources, serviceability maximization for reliable services should be facilitated by using stable cloud computing infrastructure, and resource consumption minimization for specific group-based services should be attracted to a localized P2P computing platform [5], [6].

In this article, we consider the cloudization framework for next-generation mobile computing, wherein the utilization of multitiered computing resources is driven by diverse optimization objectives. First, we analyze a cloudization architecture that adopts the network functions virtualization management and orchestration (NFV-MANO) specifications [7] standardized by the European Telecommunications Standards Institute (ETSI). Next, the state-of-the-art effective approaches are reviewed following an optimization objective taxonomy. In addition, intrinsic computing is discussed as an extension of the cloudization framework. Finally, we highlight the current issues and research directions.

II. Driving Factors for Next-Generation Mobile Computing

Next-generation mobile networks are expected to provide diverse services in three intended usage scenarios: enhanced mobile broadband (eMBB), ultrareliable and low-latency communications (URLLC), and massive machine-type communications (mMTC), as identified by the International Telecommunications Union – Radiocommunication (ITU-R) sector [8]. These scenarios result in the following factors that drive the development of next-generation mobile computing.

Quality of experience: eMBB aims at facilitating human-centric services for access to multimedia content such as video streaming, augmented reality (AR) applications, and highly interactive online games. These services prioritize user satisfaction, which is measured by the quality of experience (QoE) metric. From the perspective of the influence of mobile computing, the required QoE criteria are mainly characterized by ultralow latency and high performances for complex-data handling and large-data storage.

Precision operation: Precision operation predominantly arises in URLLC, where stringent requirements for sustainable mobile computing capabilities in terms of the availability,

The authors are with Chung-Ang University, School of Computer Science and Engineering, Seoul, Republic of Korea.
Corresponding author: S. Cho (srcho@cau.ac.kr)

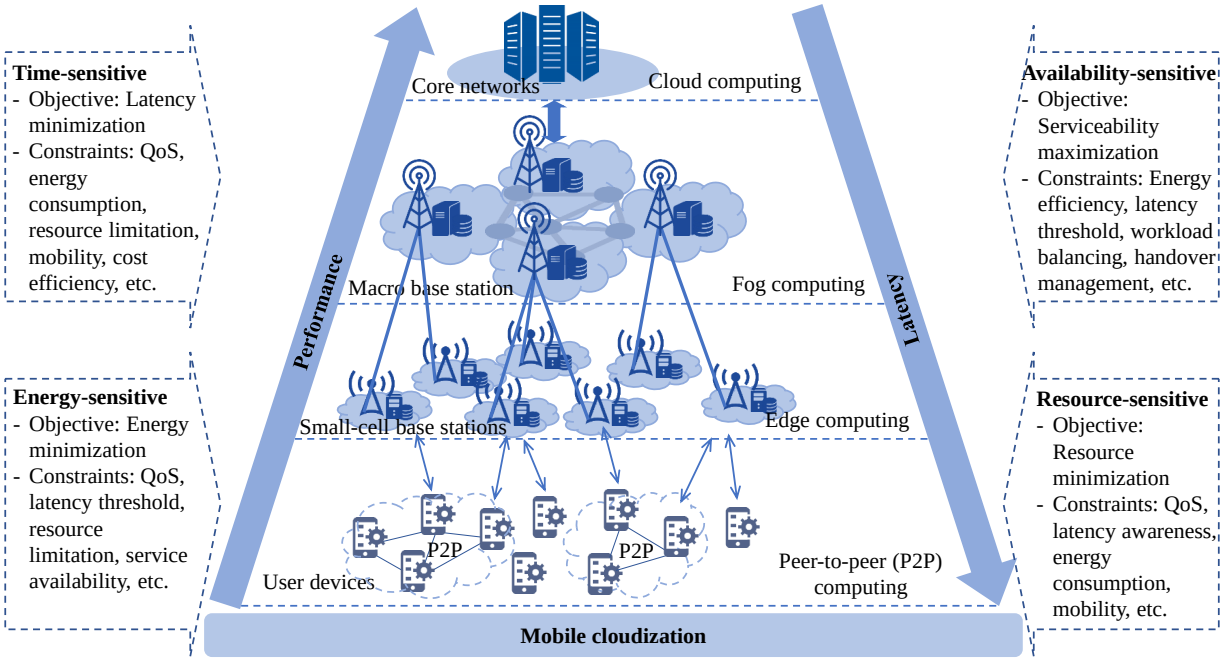

Fig. 1. Mobile cloudization model and objectives.

reliability, and energy efficiency exist in addition to a mandatory low latency. Among a broad variety of applications, smart manufacturing, remote medical surgery, mission-critical systems, and self-driving cars are prime examples. It is observed that such URLLC-enabled applications operate in fault-sensitive domains.

IoTization: The IoTization paradigm represents the emergence of connected things for smart living, where these things are interconnected via digital ecosystems in order to accommodate humans in the most convenient manner, e.g., smart cities, smart homes/buildings, etc. These use cases are known to be mMTC applications that are characterized by heterogeneity, mobility, and massiveness. Although an individual mMTC-specified thing typically generates low-volume traffic, providing stable mobile computing services for a very large school of such things with diverse application-specific requirements is a very large challenge. In addition, the energy efficiency must be considered as a further requirement to maintain communications with respect to battery constraints.

Resource efficiency: Optimal resource utilization plays an important role in almost all communication systems. Regarding mobile computing, there are two perspectives that must be taken into account to obtain this achievement: user devices and network operators. The limitations of the computing capability and the workload reduction strategy of user devices, especially lightweight mMTC-specified things, increasingly affect the offloading decision to mitigate service executions from the devices to the networks. As a result, the mobile computing infrastructure, in turn, bears a tsunami of these computing requests. Therefore, resource efficiency is a crucial factor to handle these requests since the mobile computing infrastructure itself is faced with a limited resource capacity.

Cost reduction: Last but not least, the operating cost significantly affects the final offloading decisions and optimization strategies made by the user devices and network operators, respectively. In particular, the user devices desire to obtain the best QoE services required within a reasonable cost. Meanwhile the network operators do their best to accommodate the user requirements with a maximum cost reduction.

In summary, these aforementioned driving factors lead to four foundational optimization approaches for the mobile computing infrastructure: (i) latency minimization for time-sensitive services, (ii) energy minimization for green and battery-limited platforms, (iii) resource minimization for resource-constrained operations, and (iv) serviceability maximization for massive and reliable applications. Depending on particular requirements, these objectives can be considered to develop either a joint or standalone optimization function subject to multiple constraints.

III. Mobile Cloudization

The ETSI MANO [7] defines a reference architecture for NFV management and orchestration purposes. In particular, MANO provides configuration and provisioning facilities for integrating, managing, and maintaining the correct operations of NFV infrastructures. From the cloudization functions' perspective, Fig. 2 depicts the cloudization architecture adopting the ETSI MANO specifications. Accordingly, an orchestrator acts as the central entity that incorporates four virtualized infrastructure managers (VIMs) via the `Or-Vi` main reference points. Each VIM handles both hardware and virtual resources as well as their virtualization in a corresponding computing tier. The `Nf-Vi` main reference points are used for these interactions. In addition, the `VI-Ha` execution reference points connect the hardware resources and the virtualization layer. For instance, software-defined networking (SDN) can be

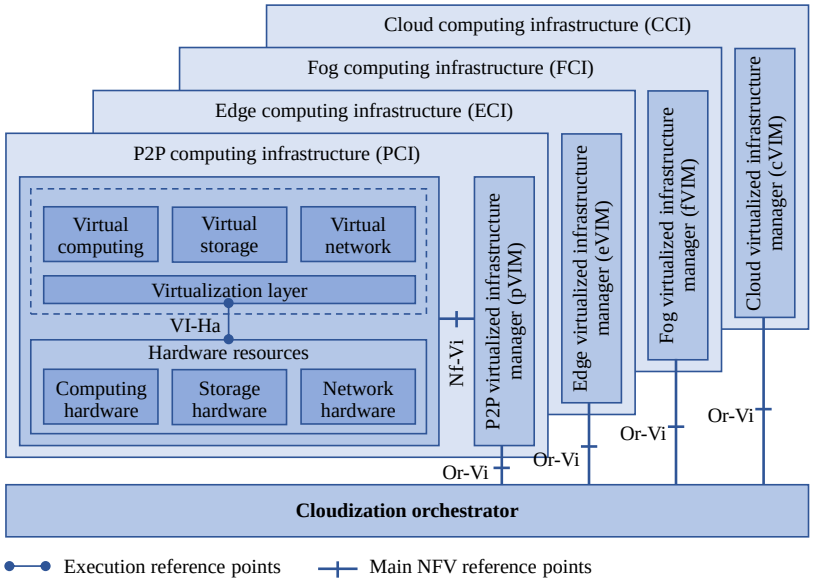

Fig. 2. Mobile cloudization architecture adopting ETSI standard.

utilized for virtualization, wherein SDN controllers abstract physical hardware elements to provide a unique virtual infrastructure in each computing tier. In the control plane, SDN controllers obtain commands from VIMs, which operate under the harmonization of the cloudization orchestrator.

A. Architectural Components

Although the computing tiers are differentiated from each other in terms of capability and performance, they provide the same services. Therefore, we clarify the functionalities of the central orchestrator for the entire cloudization architecture and two entities including the VIM and computing infrastructure, which are general for every computing tier.

1) Central Orchestrator: The central orchestrator aims at harmonizing operations among VIMs to obtain the optimal objective. To this end, the orchestrator performs three functions: workload migration across computing tiers, flexible resource utilization, and execution scheduling. The first decides *what* arrived user services are assigned to which computing tier (*where*), while the second addresses the questions *how many* and *which* particular resources should be used for the assigned workload. Finally, the third makes a plan as to *when* the assigned workload should be executed. All of these functions are driven by the optimization strategies for the optimal objective. The central orchestrator obtains the whole status picture of computing resources through communicating with the VIMs. In other words, the central orchestrator controls the resource utilization at a high level instead of direct handling.

2) VIMs: The VIMs manage both hardware and virtual resources in a single computing tier. In principle, multiple VIMs can be implemented according to each type of resource (i.e., computing, storage, and networking), as standardized in the ETSI GS NFV-MAN specification [7]. However, a unified VIM is applied for each computing tier in the cloudization architecture because these three resources are jointly considered in order to achieve the optimal objective. The VIM obtains the resource status and handles resource utilization through local resource management platforms. Specialized for each computing tier, the VIMs are named cVIM, fVIM, eVIM, and pVIM for the cloud, fog, edge, and P2P tiers, respectively.

3) Computing Infrastructures: A computing infrastructure encompasses all of the hardware and virtual resources (e.g., computing, storage, and networking) in a particular computing tier as well as their resource management platforms. If a virtualization feature is available, the resource management platforms are equivalent to the network controller and/or hypervisor components (e.g., the SDN controller and Hyper-V system), represented as the virtualization layer in Fig. 2. The virtualization layer abstracts physical hardware resources to provide virtual functional components that have capabilities and performance tailored on demand. On the contrary, if a virtualization feature is unavailable (typically in P2P computing infrastructures), the resource manager of the operating system (OS) in computing-shareable devices is in charge of the resource management platforms' responsibility. Localized for each computing tier, the computing infrastructures are referred to as CCI, FCI, ECI, and PCI in the cloud, fog, edge, and P2P tiers, respectively.

B. Capability and Performance

The cloudization architecture is able to flexibly support on-demand services to user devices since the four computing tiers provide broad ranges of computing capabilities and performance corresponding their dedicated locations shown in Fig. 1. Briefly, concentrating on specific application sectors, each computing tier is characterized as below:

- Cloud computing is deployed in data centers to provide the highest capability; however it sustains the longest latency.

- In contrast, fog computing is located in macro base stations to handle interdomain traffic with an intermediate capability and latency.
- On the other hand, edge computing involves the collaboration among small-cell base stations to allow several offloading services in close proximity to user devices.
- Lastly, P2P computing is provided on the basis of the shared resources among user devices for local computing services with an ultralow latency.

IV. Optimization Approaches

In this section, we classify the objectives of cloudization utilization into four main categories: latency, energy, serviceability, and resource optimization. Table I summaries a comparison among these categories. Regardless of the objectives, the optimization solution must address the question chain of workload assignment, resource allocation, and operation scheduling, as indicated in the central orchestrator's description section; that is, *"**What** arrived user services are assigned to which computing tier (**where**)?, **How many** and **which** particular resources should be used for the assigned workload?, and **When** should the assigned workload be executed?"*.

A. Objectives

1) Latency Optimization: The computing latency defines the amount of time a service takes to traverse the cloudization framework. In particular, if the endpoint of the service is inside the cloudization framework, the computing latency expresses how much time it takes to reach the endpoint. Otherwise, if the endpoint is outside the framework, the computing latency is determined by the duration from the time point at which the service enters the framework to the time point at which the service leaves the framework. Generally, the computing latency consists of transmission, buffering, and execution times.

From an optimization perspective, it is desirable to minimize the computing latency. To this end, the P2P and edge computing tiers are mainly utilized to reduce the transmission and buffering latencies. In addition, the quality of service (QoS), energy consumption, resource limitations, mobility, and cost efficiency are considered as key constraints to develop the optimal solution. Mostly, latency minimization benefits user services such as smart manufacturing, mission critical systems, remote medical surgery, and self-driving cars.

2) Energy Optimization: The computing energy consumption is the amount of energy used to compute a service in the cloudization framework. The computing energy is consumed during the transmission, buffering, execution, and storage processes. However, the buffering and storage energies are mostly excluded when calculating the optimization solution since they are considered as constant values that are simply determined on the basis of the amount of data.

The objective is to minimize the computing energy consumption. It is worth noting that the energy efficiency is one of the most important criteria for next-generation mobile networks towards a green ecosystem. Among the four computing tiers in the cloudization framework, P2P computing concerns the battery limitations of user devices, while fog computing and cloud computing aim at reducing the energy owing to their large data handling responsibility. The main constraints for energy minimization include the QoS and latency threshold of the offloaded services, the resource limitations of the computing infrastructures, and the service availability guarantee. Finally, energy optimization benefits cloudization providers with regard to the operational cost while maintaining the requirements of user services.

3) Serviceability Optimization: The computing serviceability is defined as the ability of the cloudization framework to serve offloaded user services within their desired requirements (e.g., latency, data volume, concurrent sessions, and service drop rate) [9]. To be more specific, the serviceability is calculated by the percentage of the number of successfully executed services per the cumulative number of offloaded services during a certain time interval. For some specific purposes, the serviceability might be considered via the fairness among computing devices/tiers (a.k.a. workload balancing) or the service drop rate (i.e., the negative metric of the serviceability).

In term of optimization, it is desirable to maximize the computing serviceability. Because service adaptation is a response of network providers, edge computing, fog computing, and cloud computing which belong to the network infrastructure, are mainly optimized to maximize the serviceability. To this end, the computing infrastructures must overcome several impediments such as resource limitations, handover management, and workload balancing. The serviceability is very important to realize massive ecosystems in next-generation mobile networks, e.g., smart cities, smart homes/buildings, and dense IoT systems.

4) Resource Optimization: The resource utilization is determined by the number of resource units that the cloudization framework uses to execute the given offloaded services. Resources related to the computing activities consist of the chip frequency (unit: Hz), memory (unit: bytes), storage (unit: bytes), and hauling bandwidth (unit: resource blocks). Note that other resources such as the energy, time, and space are beyond the scope of this optimization.

Towards a resource-efficient environment, the cloudization framework should optimize the resource utilization at minimum. Although resource minimization has been considered as a crucial characteristic for every efficient computing infrastructure, the P2P and edge computing tiers in the cloudization framework strictly prioritize this objective owing to their limited resource powers. In addition, the QoS, latency awareness, and user devices' mobility are also the main constraints for this optimization. Potential systems and services that benefit from resource optimization include lightweight devices, dense IoT systems, multimedia services, etc.

5) Joint Optimization: In order to improve the applicability of these strategies in real scenarios, joint optimization (a.k.a. hybrid or balancing solutions) is typically considered [10]. The mathematical expression of the joint optimization is formed to minimize or maximize a summation of multiplications of the objective functions by their coefficient factors. The coefficient factors prioritize their objective functions; and they are in range of [0, 1.0]. In other words, joint optimization can be

TABLE I
OPTIMIZATION OBJECTIVES

Characteristics	**Latency optimization**	**Energy optimization**	**Serviceability optimization**	**Resource optimization**
Objective	Minimization	Minimization	Maximization	Minimization
Components	Transmission, buffering, and execution	Transmission and execution	Offloaded services	Hauling bandwidth, CPU, memory, and storage
Main targets	P2P and edge	P2P, fog, and cloud	Edge, fog, and cloud	P2P and edge
Constraints	QoS, energy consumption, resource limitation, mobility, cost efficiency, etc.	QoS, latency threshold, resource limitation, service availability, etc.	Energy efficiency, resource limitation, workload balancing, handover management, etc.	QoS, latency awareness, energy consumption, mobility, etc.
Beneficiaries	Users	Providers	Users and providers	Providers
Potential systems and services	Smart manufacturing, remote medical surgery, mission-critical systems, self-driving cars, etc.	Battery-limited devices, green networks, smart grids, etc.	Smart cities, smart home, smart building, dense IoT systems, etc.	Lightweight devices, multimedia services, dense IoT systems, etc.

utilized to harmonize the optimization objectives. For instance, the coefficient factor of the latency in a joint minimization applied to the ECI should be closer to 1 while the coefficient factor of the energy can be closer to 0.

B. Solutions

The infrastructure components related to cloudization optimization consist of user devices (battery capacity, antenna specifications, mobility characteristics, and deployment density), computing devices (CPU frequency, buffer size, storage capacity, mobility characteristics, and the number of devices), and communication links between the user devices and computing devices and among the computing devices (bandwidth and quality of channels). Depending upon the particular optimization objective, mathematical expressions of the optimization utilities are developed by applying the appropriate analysis tools such as probability, queuing, graph, and game theories.

In the probability perspective, user service arrival at the cloudization framework is considered adopting a discrete probability distributions (e.g., Poisson and Zipf). In case none of the well-known distributions is appropriate, the user service arrival process might be possibly patternized using machine learning classification techniques. Once the user service arrival becomes deterministic, optimization functions can be developed accordingly. On the other hand, queuing-theoretic approaches consider the computing process in the cloudization framework as a queuing system. The arrival and service processes of the queuing system are characterized based on the user service arrival and cloudization computing capacity, respectively. In contrast, graph-theoretic approaches transfer the system into a bipartite model, where vertices and links present the computing devices/user devices and communication channels, respectively. This approach mainly focuses on addressing the computing problem regarding communication channels (e.g., unstable transmission environment and insufficient bandwidth) and service assignments from the user devices to computing devices. Last, game-based approaches consider all the infrastructure components as players. Since the cloudization framework is centrally managed by the orchestrator, the information, behavior, and actions of the players are assumed to be known. Consequently, the optimization function is developed adopting the equilibrium strategy.

The final objective functions should be transformed or relaxed into a well-known type of convex/nonconvex and/or linear/nonlinear problem. Typically, the problems might be resolved by using finite-step algorithms, iterative methods, or heuristics. Detailed classifications and solutions are provided in [11].

V. INTRINSIC COMPUTING

While cloudization is receiving a considerable amount of attention to improve its performance to accommodate the emergence of service offloading from a massive number of IoT devices, another computing trend is also burgeoning on the user-device side, namely intrinsic computing. As the hardware performance of user devices significantly increases every year, the intrinsic computing strategy illustrates the ability of user devices to cooperate using heterogeneous computing components such as central processing units (CPUs), graphics processing units (GPUs), and field-programmable gate arrays (FPGAs) to execute user services [12]. To realize this, an open computing language (referred to as OpenCL) [13] has been proposed by the Khronos Group to enable a uniform environment for general-purpose parallel programming across these computing components.

Fig. 3 depicts a typical intrinsic computing platform based on the OpenCL model. The platform defines a computing environment that consists of a *host* (e.g., a CPU) connected to one or more *compute devices* (i.e., the remainder of the aforementioned computing components). A compute device is divided into multiple compute units (CUs), which are, in turn, further divided into a number of processing elements (PEs). Accordingly, several set of memory are assigned for these computing levels. To perform a user service, the service must be implemented as both *host code* and *kernel code*. The

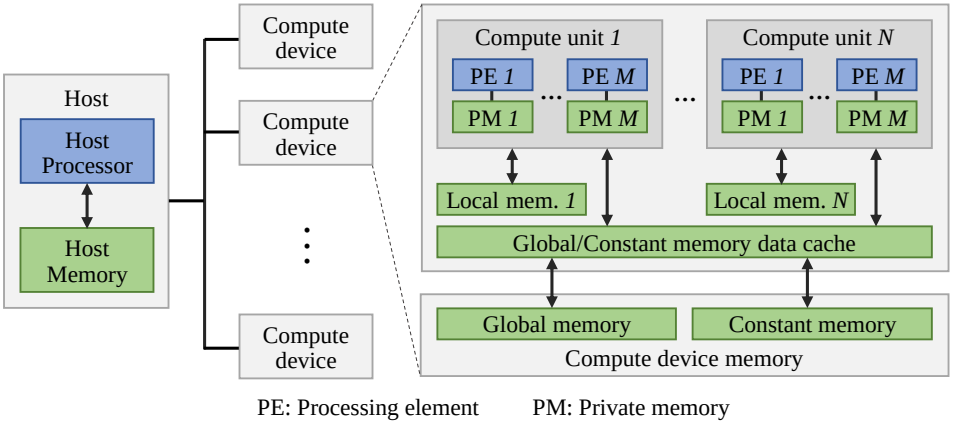

Fig. 3. OpenCL-based intrinsic computing platform.

host code is run by the host processor adopting the native regulations of the hardware platform. The host code handles the service computations by submitting kernel codes as commands to the computing devices. Finally, the computations are executed within the PEs.

Intrinsic computing is applicable for high-performance user devices such as modern smartphones, tablets, and laptops. Under these circumstances, the user devices are able to decide whether their services should be (partially) offloaded to the cloudization framework or internally executed using intrinsic computing.

A. Self-optimization Perspective

Self-optimization aims to benefit intrinsic computing with energy efficiency and resource minimization since user devices have limited energy and resource capacities. A comprehensive literature review [14] showed that energy-efficient resource utilization and workload scheduling in intrinsic computing can obtain optimal performance by using numerous techniques such as dynamic voltage/frequency scaling (DVFS), workload balancing, and service-specific awareness. Among these techniques, DVFS dynamically controls the chip frequencies and operational voltage in order to minimize the energy consumption with strict consideration of consequent processing-performance decrease. Workload balancing manages the service computation assignment among compute devices/units for optimal resource utilization. In addition, workload balancing schemes deactivate the compute devices/units that are not used for service executions to reduce the energy consumed. On the other hand, service-specific-aware schemes consider the characteristics of the services to assign service computations to the appropriate compute devices. For instance, GPUs are good at matrix operations and image processing, while general operations should be processed by CPUs for better performance.

B. Offloading Decision

Since user devices are consumers of cloudization services, user devices have full rights to make their offloading decisions, which means using either the cloudization services or their own intrinsic computing. The offloading decisions can be full offload, partial offload, or no offload. The decisions depend on the service demands in terms of the latency, energy, resource, and/or cost efficiencies with joint consideration of the beneficial offers between intrinsic computing and cloudization computing. It is worth noting that the wireless channels connecting user devices and networks have significant impacts on the beneficial offers of cloudization computing. The detailed classification of offloading decisions is described in [10].

VI. Research Directions

Contextual adaptation: From a computing perspective, context consists of the computational environment states and settings that reveal the offloaded services' characteristics (e.g., service type and device location) and the computing infrastructures' conditions (e.g., resource states, communication link quality, and current workload) during service execution. Contextual adaptation enables the autoreconfiguration ability for cloudization framework switching among optimization strategies. For instance, resource minimization should be prioritized during rush hours, while serviceability maximization should be activated for serving high-mobility devices.

Algorithmic complexity reduction: Currently, one of the native challenges that optimization solutions face is a high algorithmic complexity because almost all of the optimization functions are identified as nondeterministic polynomial-time (NP)-hard problems. The problems become more severe for large-scale optimization, as in the cloudization framework. To be more specific, the algorithmic complexity consists of time and space aspects. The time complexity is concerned with how long it takes to perform the optimization algorithm. The time complexity is measured by the number of elementary operations. On the other hand, the space complexity specifies how many bytes of memory are occupied by the optimization algorithm to find the results.

Elasticity and scalability: Since the cloudization framework has a tiered architecture, optimization solutions should work elastically either in each computing infrastructure or in the entire framework depending on user service demands. In addition, scalability is also necessary to handle the rapid growth of big data offloaded from the massive number of IoT

devices. Although the cloudization framework has been designed adopting the rule of *centralized control and distributed operation*, flexible cooperation among the computing portions still requires further improvement.

Security and privacy: The cloudization framework directly manipulates user information that is stringently sensitive to security and privacy issues [15]. On the framework side, the vulnerabilities of system software and denial of service (DoS) attacks are open challenges. Moreover, data loss and inadequate data backups possibly lead to privacy violations, especially in P2P and edge computing infrastructures because of storage resource limitations. On the other hand, most IoT devices require lightweight security protocols owing to their insufficient performance. This circumstance makes the communications between user devices and the framework vulnerable against eavesdropping attacks.

VII. Concluding Remarks

This article presented an overview of mobile cloudization regarding a standardized framework architecture and features, the optimization objectives and effective approaches for framework utilization, intrinsic computing beyond the infrastructure-based cloudization, and research directions for current issues. Efficient cloudization could result in a broader range of applicable services. Although the cloudization framework has provided a convenient cloudization infrastructure for service offloading in next-generation mobile networks, several technical challenges still need to be resolved in order to obtain optimal operation, especially in the dynamic heterogeneous IoT paradigm.

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," White Paper, Cisco, February 2017.

[2] S. Dustdar, C. Avasalcai, and I. Murturi, "Edge and Fog Computing: Vision and Research Challenges," in *Proc. of IEEE International Conference on Service-Oriented System Engineering (SOSE)*, San Francisco East Bay, CA, USA, April 4–6, 2019, pp. 96–105.

[3] M. Gusev and S. Dustdar, "Going back to the roots - the evolution of edge computing, an IoT perspective," *IEEE Internet Computing*, vol. 22, no. 2, pp. 5–15, 2018.

[4] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[5] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2018.

[6] M. Gusev, B. Koteska, M. Kostoska, B. Jakimovski, S. Dustdar, O. Scekic, T. Rausch, S. Nastic, S. Ristov, and T. Fahringer, "A deviceless edge computing approach for streaming IoT applications," *IEEE Internet Computing*, vol. 23, no. 1, pp. 37–45, 2019.

[7] *Network functions virtualisation (NFV); Management and orchestration*, ETSI Std. GS NFV-MAN 001 V1.1.1, December 2014.

[8] "Minimum requirements related to technical performance for IMT-2020 radio interface(s)," Report M.2410-0, ITU-R, November 2017.

[9] N.-N. Dao, J. Lee, D.-N. Vu, J. Paek, J. Kim, S. Cho, K.-S. Chung, and C. Keum, "Adaptive resource balancing for serviceability maximization in fog radio access networks," *IEEE Access*, vol. 5, pp. 14 548–14 559, 2017.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[11] R. L. Rardin, *Optimization in operations research*. Prentice Hall, 2016.

[12] M. Halpern, Y. Zhu, and V. J. Reddi, "Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction," in *Proc. of IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Barcelona, Spain, March 12–16, 2016, pp. 64–76.

[13] A. Bourd, *The OpenCL specification v2.2-3*, Khronos Group, May 2017.

[14] S. Mittal and J. S. Vetter, "A survey of CPU-GPU heterogeneous computing techniques," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 69, 2015.

[15] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.