Received August 5, 2019, accepted August 12, 2019, date of publication August 30, 2019, date of current version September 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938660

Toward Computation Offloading in Edge Computing: A Survey

CONGFENG JIANG[1,2], (Member, IEEE), XIAOLAN CHENG[1,2], HONGHAO GAO[3], (Member, IEEE), XIN ZHOU[1,2], AND JIAN WAN[2,4]

[1]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China
[3]Computing Center, Shanghai University, Shanghai 200444, China
[4]School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Corresponding author: Honghao Gao (gaohonghao@shu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61972118 and Grant 61972358, and in part by the Key Research and Development Program of Zhejiang Province under Grant 2018C01098, Grant 2019C01059, and Grant 2019C03134.

ABSTRACT The explosive growth of massive data generation from Internet of Things in industrial, agricultural and scientific communities has led to a rapid increase for data analytics in cloud data centers. The ubiquitous and pervasive demand for near-data processing urges the edge computing paradigm in recent years. Edge computing is promising for less network backbone bandwidth usage and thus less data center side processing pressure, as well as enhanced service responsiveness and data privacy protection. Computation offloading plays a crucial role in edge computing in terms of network packets transmission and system responsiveness through dynamic task partitioning between cloud data centers and edge servers and edge devices. In this paper a thorough literature review is conducted to reveal the state-of-the-art of computation offloading in edge computing. Various aspects of computation offloading, including energy consumption minimization, Quality of Services guarantee, and Quality of Experiences enhancement are surveyed. Moreover, resource scheduling approaches, gaming and tradeoffing among system performance and overheads for computation offloading decision making are also reviewed.

INDEX TERMS Edge computing, computation offloading, task partitioning, game theory, edge-cloud collaboration.

I. INTRODUCTION

The cloud computing paradigm is a service provisioning model that provides user access to scalable distributed capabilities including computing, networking, and storage in the cloud data centers. Cloud service providers (CSPs) provide flexibility and efficiency for end users by providing services such as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). For example, service vendors can scale services to fit their needs, customize applications and access cloud services from anywhere with an internet connection. Thus cloud-based services are ideal for businesses with growing or fluctuating internet bandwidth demands. Moreover, with cloud computing, enterprise users can ship applications to market more quickly, without worrying about underlying infrastructure costs, maintenance, disaster recovery, and automatic software updates. To leverage the benefits of cloud computing, various deployment models including private cloud, public cloud, and hybrid cloud, are key factors for system reliability and scale for business needs. From its introduction, cloud computing has changed the way of business of all vertical domains as well as human being's daily life dramatically. Furthermore, enterprise IT investments for cloud-based offerings will be faster than growth in traditional (non-cloud) IT offerings, such cloud shift from traditional software orientation making cloud computing one of the most disruptive forces in IT markets. Therefore, it's predicted that cloud service is inevitably becoming pervasive and ubiquitous in any commercial or personal market, which is similar with the prevalent dominance of Internet in nowadays.

Varghese and Buyya [1] review the evolution and advance of cloud computing from cloudlet, ad hoc cloud, multi-cloud, heterogeneous cloud micro-cloud, and introduce four

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Li.

emerging cloud computing architectures including fog and mobile edge computing [2], volunteer computing, serverless computing and software-defined computing. They also discuss the future impact of cloud computing on IoT (Internet of Things), big data, and autonomous learning systems and identify the challenges of developing the next cloud computing system including security and reliability enhancement, sustainable cloud infrastructure, and efficient resource management strategies.

Although cloud computing can provide organizations dynamic, cloud-based operating models for cost optimization and increased competitiveness, it also has some disadvantages in many scenarios like industrial IoT, connected autonomous vehicles (CAVs), smart homes, and smart cities. For example, cloud computing based processing requires huge volume of data transportation from end devices and sensors, which consumes large network bandwidth. Moreover, cloud data center based analysis is not possible for huge data generated from thousands of millions of end devices due to the incapability of computing and storage. Therefore, cloud computing based processing can't provide prompt responsiveness and short latency for big data analytics from massive IoT devices. Moreover, in some scenarios where data privacy and security is the first concern, cloud computing data centers are not trustful to conduct the data analytics. In contrast, data privacy preserving requires that the data is processed near its source, other than in the remote cloud data centers.

Edge computing [3] is emerged as a promising paradigm that provides capabilities of processing or storing critical data locally and pushing all received data to a central data center or cloud storage repository. For example, in IoT use cases, the edge devices collect data from sensors and process it there, or send it back to a data center or the cloud for processing if the local processing power is not enough. To this end, edge computing paradigm can take some of the load off the central cloud data centers and migrate the tasks from cloud computing centers to network edge devices, reducing or even eliminating the processing workload at the central location. Similarly, Fog computing mitigates the potential of IoT services and new resource sharing as a complement of traditional cloud computing models and the combination of fog-cloud, and fog-cloud integration provides a foundation for creating a new highly heterogeneous computing and network architecture [4], [5].

The demand for scalable real-time data analytics in IoT scenarios is the main driving force for edge computing. In edge computing environment, data generation and consumption are concentrated to the edge of the network in many applications of smart home, smart city, and industrial internet. In edge computing, computation offloading plays an important role in latency minimization and Quality of Services guarantee. Specifically, in order to tradeoff among system overheads, energy consumption, and system performance, tasks may be offloaded to edge devices from the cloud data centers. Various computation offloading strategies and approaches [6]–[37] are proposed, including game and cooperation between edge and cloud, heuristic offloading, etc. In addition, computation offloading oriented optimization is also proposed, such as the collaboration between the edge and the cloud [38]–[41], and energy-efficient computation offloading and resource allocation [42]–[52].

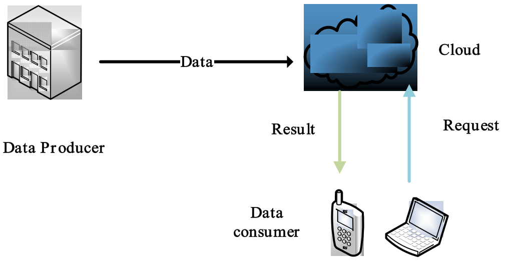


FIGURE 1. The cloud computing model.

The reminder of this paper is organized as follows. In Section 2, we introduce the basic concepts of edge computing. Then we survey some work on edge-cloud collaboration for computation offloading in section 3. We evaluate the work on decision making of computation offloading in section 4. The case studies of computing offloading strategies are selected and discussed in section 5. In section 6, we give some review on performance evaluation and simulation of edge computing. Finally, we summarize the paper in section 7.

II. THE EDGE COMPUTING PARADIGM

A. EDGE COMPUTING ARCHITECTURE

Edge computing is emerging as a new computing model where the processing to data is close to the date source, i.e., the network edge. In this scenario, the network edge can be places where data storage, computing, or networking services is performed. With the advances of IoT, 5G communication, autonomous driving, and smart cities, edge computing is connecting and bridging the gap between numerous end devices and the centralized cloud computing data centers. As edge computing tries to bring application hosting from centralized data centers down to the network edge, closer to end users and the data generated by applications, it can improve content delivery and application user experience by shortening network transmission path between the end user's device and the location where the data they are accessing is placed. Moreover, in some cases where data privacy and security is the main concern, edge computing promises to provide data privacy preservation by keeping data inside the network edge rather than sending the data to centralized cloud data centers, which in turn provides lower latency, increase reliability and improves overall network efficiency.

Figure 1 gives the cloud computing model, where the data producers transmit the generated source data to the cloud while the terminal devices such as servers, personal computers, mobile phones, and other devices send requests to the cloud center to obtain data processing results.

Since the cloud computing paradigm is designed for centralized service provisioning based on the economy of scale,

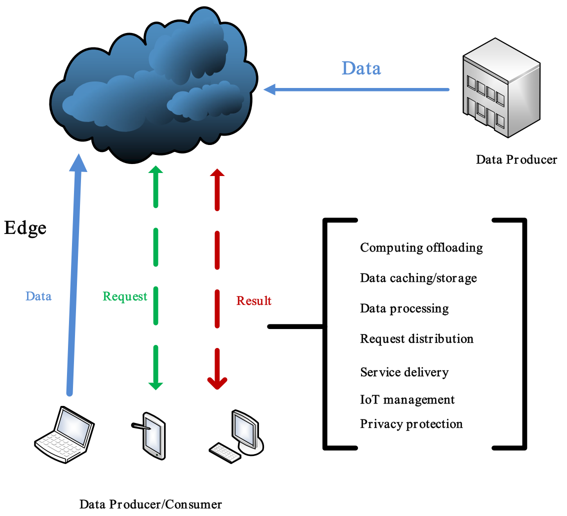


FIGURE 2. The edge computing model [57].

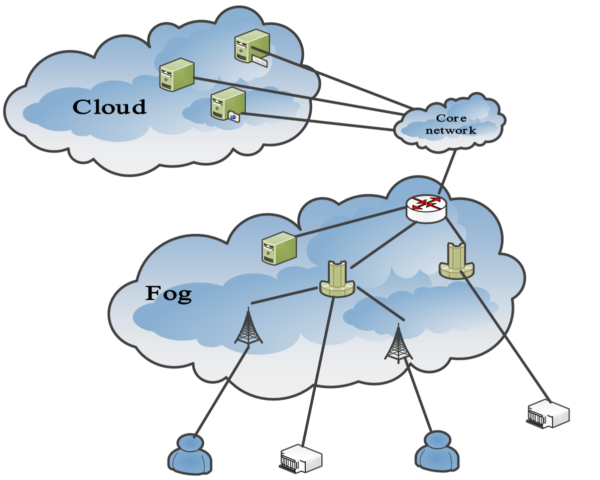


FIGURE 3. Collaboration between edge and fog computing.

TABLE 1. Existing work on the edge computing paradigm.

Work	Contributions
Varghese *et al.* [1]	Cloud computing evolution
Ramirez *et al.*[4]	F2C(Fog-to-Cloud)
Masip-Bruin *et al.* [5]	OpenFog RA, F2C
Shi *et al.* [3][53][54]	Vision and case studies
Li *et al.*[55] Zhang *et al.* [56]	Programming Model

it's not suitable and capable to provide processing capabilities on numerous decentralized edge devices due to constraints of backbone network bandwidth and processing power in cloud data centers.

Moreover, since hundreds of millions of edge devices are geographically deployed in a distributed manner, and the processing to data is also performed on heterogeneous distributed devices, it's very important to design new system architecture suitable for edge computing environments, because the data volume generated by various applications and devices running in the edge computing environment is huge and highly heterogeneous. Figure 2 presents the edge computing model.

With the ever-increasing deployment of various IoT devices, lots of mobile devices and applications needs more stringent requirements on service quality and real-time responsiveness of data processing. Unlike the traditional cloud computing model, in the edge computing model all computations and processing are performed at the edge of the network and it extends computing, networking, and storage capabilities from the cloud data center to the edge of the network, to fully exploit the computing power of end edge devices. Shi *et al.* [3] presents the challenges of reliability, isolation, scalability, and differentiation that may be faced when designing new architectures and operating systems in edge computing environments and the corresponding solution through the studying of the edge operating system architecture deployed in smart homes.

In order to adapt to various application scenarios and meet different service requirements in the cloud computing environment, various modified computing models have been proposed. The Fog computing model was proposed and defined as a highly virtualized computing platform for migrating cloud computing center tasks to network edge devices. As shown in Figure 3, fog computing centralizes data storage, processing, and applications into devices on the network edge, eliminates the need to save all of the data to the cloud data center and adds an intermediate layer between the end device and the data center.

The intermediate layer consists of fog servers deployed at the network edge to reduce the communication between cloud data centers and edge devices and reduce the bandwidth usage and power consumption of the backbone networks. Moreover, the cloud-edge collaboration generates high network communications and service delays while the data privacy and energy consumption during data transmission are also should be considered. Shi *et al.* [53] argue that edge computing and fog computing have some similarities, but the difference is that the fog computing focuses on the management of back-end distributed shared resources, while edge computing emphasizes the design and implementation of edge intelligence in addition to the infrastructure and edge devices, extends the processing power to the end devices, and the real-time processing of data is done by devices in the edge network. We list some research work on the basic concepts of edge computing, mobile edge computing, and fog computing in Table 1.

B. RESOURCE ABSTRACTION AND PROGRAMMING MODEL

In cloud computing environment, users of cloud services can write and compile code on the target platform and then run it on the cloud server without prior knowledge of the deployed infrastructure. However, in the edge computing environment, the task execution model is different from that in cloud computing. The tasks in edge computing can be partitioned into several subtasks and each subtask can be offloaded to different edge devices for faster execution. In order to provide low latency of parallel subtask execution, it's required that the task must be partitionable and migratable for data processing on the edge devices.

Ramírez *et al.* [4] evaluated the potential benefits of Fog-to-Cloud (F2C) architecture in dynamic service scenarios, including service's response time, power consumption, network bandwidth usage, and the probability of service outage. Their results show that the combined fog-to-cloud (F2C) architecture brings significant performance benefits compared to the traditional pure cloud computing based solution. Masip-Bruin *et al.* [5] compared two existing hierarchical resource architecture models, i.e., OpenFog RA and F2C, on resource continuity and collaborative management, and proposed a distributed management framework which is effective to guarantee resource continuity within a layered architecture.

In order to maximize the performance of edge computing based applications, Li *et al.* [55] designed a lightweight programming language, namely, *EveryLite*. Their experimental results show that the execution time of EveryLite is lower 77% and 74%, and the memory footprint is 18.9% and 1.4%, comparing to *JerryScript* and *Lua*, respectively.

In the forthcoming era of Internet of Everything (IoE), edge devices can act as both data producer and consumer, which makes it possible to process private data close to data owner. Zhang *et al.* [56] proposed the *Firework* programming model for edge computing, which contains *Firework Manager* and*Firework Node*. The *Firework* model combines geographically distributed data sources by creating virtual shared data views, while stakeholders, i.e., the *Firework Nodes*, provide end users with a set of predefined functional interfaces for user access. The interface of the *Firework* model is a set of data sets and functions, and the functions are bound to the data. To this end, the *Firework* model makes the data processing closer to the data producer and reduces the response delay. Moreover, since all data stakeholders in the *Firework* model need to register their corresponding data sets and responsive functions into a data view and the registered data views are visible to all participants in the same *Firework* model, any participants can combine multiple data views to conduct data analysis in a specific context. The *Firework Manager* decomposes the service request into several subtasks that are sent to each participant, and then each participant will perform the corresponding computing task on their local device.

C. CASE STUDIES OF EDGE COMPUTING APPLICATION

Application cases are the most direct and effective way to verify whether new technologies are valuable, which is also reliable when it comes to edge computing. Nowadays, edge computing has been applied in IoT, smart home, intelligent transportation, and smart city [54]. Chabas *et al.* [57] identified 11 industrial fields and more than 100 edge computing use cases, and argues that these industry applications can create more than $200 billion in hardware value over the next five to seven years until 2025. Three popular application cases are listed as follows:

1) IoT

The wide deployment of IoT devices and the increasing commercial demand for real-time data processing and the high quality of service of user experiences urge the creation of edge computing. Since more and more intelligent devices and sensors are deployed in the IoT environment, data production and consumption are performed and shifted to the edge of network gradually, which also needs elaborate computing technology for real time analytics and pervasive processing.

2) SMART HOME

Home is one of the most important places that each person spent more time there. Technologies for better quality life and living conditions have changed human being's lifestyle and lift quality. Deploying various sensors at home and sending collected data to remote cloud data center for processing introduce high risk of private data leakage, data abuse, and physical threat to massive local residents. Therefore, the traditional cloud computing based data processing is not suitable for smart home applications, and data privacy preservation enabled edge computing emerges as the perfect alternative to smart home.

3) SMART CITY

City is the place that consists of many smart homes, which implies that the edge computing paradigm can be extended from family level to city level, i.e., the smart city. However, since a typical city also produces large volume of public services related data, even the most advanced cloud data centers can't process these data in real time for city-scale interactive analytics due to the lack of capabilities of computing, storage, and networking. If data processing can be offloaded to the edge of the network, it can reduce the pressure of cloud data centers and make it possible for near real time analytics. Moreover, in a smart city, one of the most important application scenarios is intelligent transportation. Networked traffic sensors and cameras provide perfect platform for edge data processing close to local data source, which makes it possible to solve traffic problems facing the urban residents, from traffic conditions alerts to road conditions prediction.

III. EDGE-CLOUD COLLABORATION

Although the traditional cloud computing technology cannot meet requirements in terms of real-time response, privacy protection, and less energy consumption, the edge computing paradigm is not in essence replacing the cloud computing technology. In contrast, the cloud computing and edge computing are complementary and mutually reinforcing each other in many scenarios. Moreover, the edge computing and cloud computing will collaborate in the networked computing environment including scenarios such as IoT, smart city, smart home, industrial internet, connected autonomous vehicles, etc. The edge computing technology can fully exploit

TABLE 2. Existing work on resource allocation in edge computing.

Work	Contributions on resource allocation
Zhao *et al.* [49]	Radio and computational resources allocation
Zhang *et al.* [50]	Energy-latency, energy-aware computation offloading
Samie *et al.* [51]	Low-power IoT edge devices oriented
Zhang *et al.* [52]	Heterogeneous networks oriented
Liu *et al.* [58]	Tradeoff on energy, latency, and offloading costs
You *et al.*[48]	Energy-saving resource management strategy
Wang *et al.* [60]	*ENORM*: Resource management framework of edge node:
Tan *et al.* [61]	Resources allocation and Caching
You *et al.* [62]	Resource allocation for MECO systems
Xu *et al.* [63]	Enhanced resource management algorithm for online learning
Yang *et al.* [64]	Joint computing partition and resource allocation problem (JCPRP) solving
Liu *et al.* [65]	Bandwidth-based partitioning scheme

the computing capabilities of the edge devices perform partial or whole computing at the edge devices, and thereby reducing the computing demand of the cloud data centers and the transmission bandwidth of core network. The collaboration of edge computing and cloud computing provides more opportunities for pervasive data analytics in IoT and low latency computing for latency critical applications such as autonomous driving and industrial networked systems.

Therefore, the edge computing paradigm still needs the cloud data center's powerful computing capabilities and mass storage infrastructure, while the centralized cloud data center also needs the edge device to process the massive data on the edge devices for lower latency, privacy protection, and less energy consumption. In this section, we will elaborate on the collaboration on three aspects: resource management and allocation, execution model, and resource partitioning.

A. RESOURCES MANAGEMENT AND ALLOCATION

In decentralized edge computing environment, resource must be allocated, such as processor, disk, and network bandwidth for distributed data processing. Since edge devices may have limited resources including computing, storage, and networking I/O, resource allocation must be performed based on both existing available resources and performance constraints. Specifically, resource allocation is performed under multiple conditions, including resource usage quota, power and energy consumption budget, and latency.

We list some research work on resource allocation in edge computing environment in Table 2.

Liu *et al.* [58] tried to tradeoff between energy consumption, execution delay and offloading cost, and proposed an optimization strategy for optimizing these three objectives simultaneously. Their simulation experiments show that the joint optimization strategy can guarantee better quality of service.

Since the data arrival pattern and deadline for data processing vary significantly in different edge computing scenarios, it is not feasible to formulate general resource allocation mechanism in edge-cloud collaboration environment. Liu *et al.* [64] studied the energy-saving resource management strategy of asynchronous mobile-edge computation offloading (MECO) systems. The best data partitioning and time division policy is derived by analyzing the general arrival data series, and then the total mobile energy consumption is minimized by using the block coordinate descent approach. Some approaches [65]–[70] are proposed to help decide service selection in such scenarios to meet the real-time, privacy preservation and energy consumption minimization for big data analytics.

Similarly, Wang *et al.* [59] proposed and developed the edge node resource management framework, namely, *ENORM*. They proposed a new configuration and deployment mechanism for linking communication between edge nodes and the cloud data center such that *ENORM* can provide offloaded workloads for edge nodes. Moreover, *ENORM* integrates low overhead and dynamic auto-extension mechanism to add or remove resources to manage workloads on edge nodes effectively. They validated the feasibility of *ENORM* through context-sensitive and delay-sensitive online gaming use cases and the results show that *ENORM* can reduce application service latency up to 20% to 80% and reduce the frequency of data transmission and communication between edge nodes and the cloud up to 95%.

Currently, the networked systems are increasingly prone to be heterogeneous in terms of hardware configuration, software stack, networking media, and application domains. Specifically, data volume, data producing speed and service quality are highly diverse in edge-cloud collaboration environment. Such heterogeneity poses lots of challenges, such as how to address the shortage of mobile device resources, and how to tradeoff between the limited computing power and energy constraints of mobile nodes. Tan *et al.* [60] designed a virtual and fully duplex small scale cellular network framework based on caching heterogeneous services in edge computing. They proposed a novel resource allocation scheme that not only considers the caching mechanism, but also adopts fully duplex communication. Moreover, the proposed scheme also considers user correlation, power control, caching, computation offloading strategy, and resource allocation at the same time.

In mobile computing environment, energy consumption is the key concern for resource allocation and computing performance maximization. Researchers proposed energy saving approaches for resource allocation of single user and multiuser mobile edge computing offloading systems (MECO). However, these existing works focus on the design of complex algorithms rather than the design of optimal resource allocation strategy.

You *et al.* [61] investigated the resource allocation of multi user MECO systems based on time division multiple access (TDMA) and orthogonal frequency division multiple access (OFDMA) and consider cases with infinite or limited cloud computing capabilities. For TDMA mobile edge computing offloading systems with infinite cloud computing capabilities, they propose the resource allocation strategy by redefining the offloading priority function and modifying the previous threshold policy and then propose a low complexity sub-optimal resource allocation algorithm based on the approximate offloading priority. In other hand, for OFDMA mobile edge computing offloading systems with unlimited cloud computing capabilities, they solve the resource allocation problem as a mixed integer optimization problem and the prioritized TDMA strategy is used to optimize resource allocation, which includes: (1) translating the OFDMA resource allocation problem into a corresponding part of the TDMA, (2) determining the initial resource allocation and offloading data by defining an average offloading priority function, (3) assigning the sub channels according to the offloading order, and (4) adjusting the allocation of the offloading data on the sub channels. Simulation experiments show that this resource allocation strategy can approach optimal performance. However, the proposed approach also has some shortcomings in that they assume that: (1) the processed data can be processed separately, (2) each mobile device can perform local computation and incoming workload offloading at the same time, and (3) the edge cloud has a complete understanding of energy consumption in the local computing devices, channel gain and fairness factors of all users.

Although currently renewable energy is used to power the mobile edge computing capabilities, the intermittent and unpredictable nature of renewable energy poses a huge challenge for high quality computation offloading services. To solve this problem, Xu and Ren [62] defined this problem as a Markov decision process and proposed an efficient online resource-based reinforcement resource management algorithm, which can reduce system service latency and operating costs by real-time learning of the best strategies for dynamic job offloading and edge server provisioning. Unlike traditional reinforcement learning algorithms, the proposed online learning algorithm achieves higher learning rate and runtime performance through decomposition value iteration and reinforcement learning. The simulation results show that the system cost of the online learning algorithm is much lower than that of the compared schemes. In addition, the results also show that the proposed approach can save more power especially when the network connection is deteriorating.

B. APPLICATION COMPUTATION PARTITIONING

In edge-cloud collaboration, it's important to decide which part of the task should be offloaded to the edge devices, and which part should be offloaded to the cloud data center. Computation partitioning is the first step before computation collaboration between edge devices and cloud data centers. As the computing power of edge devices increases, applications hosted in the cloud data center can be migrated to the geographically distributed edge servers and edge nodes. Application partitioning is to decompose an application into multiple components based on state information of various aspects, including resource, power, and response delay of the edge node, while still preserving the semantics of the original application at the same time. The existing approaches include static application partitioning completed during the compilation procedure and dynamic application partitioning completed during the real time application execution. Due to the bandwidth fluctuations in wireless environments, static application partitioning is not suitable for mobile platforms with fixed bandwidth, while dynamic program partitioning will result in high overheads.

Currently, some existing work focus on partitioning applications from the mobile users' perspective, which often optimize individual mobile users to minimize the cost of execution of time or energy consumption on the device. Wang and Li [71] proposed a distributed approach that partitions the application into client side and server side, which are running on handheld devices and servers, respectively. They also construct formal analysis of constrained offloading system, which represents task mapping, data access, and data validity. Experimental results show that their solution can not only improve the performance of handheld devices, but also reduce the overall power consumption.

The traditional resource allocation formulation did not consider the network bandwidth, and some other existing work only investigated the two-dimensional resource allocation optimization, but did not provide partition decision. Yang *et al.* [63] formulated the joint computing partitioning and resource allocation problem (JCPRP) for latency sensitive applications in mobile edge cloud computing. They proposed to combine computational partitioning, edge computing resources and access bandwidth, and divide the application from the edge server's perspective rather than modeling the optimization problem from mobile users. They also designed a heuristic algorithm for multidimensional search and adjustment of resource allocation.

The communication latency between edge devices and cloud data centers is affected by various factors such as transmission distance and network bandwidth. Liu *et al.* [64] proposed a bandwidth-based partitioning scheme to improve the static partitioning performance and avoid the high cost of dynamic partitioning. Firstly, the application object relation Graphs (ORGs) is constructed by the combination of static analysis and dynamic analysis and a partitioning optimization model will be constructed. The construction of the weighted ORG is composed of the application partitioning process and the weighted object relationship graph construction. They also introduce the execution time and energy consumption of the application partitioning model, and propose three optimization models, i.e., execution time optimization, energy optimization, execution time and energy-weighted optimization. With the execution time and energy optimization partitioning model, they propose the application boundary partitioning (BBAP) algorithm based on branching boundary and the greedy application partitioning (MCGAP) algorithm based on Min-Cut. BBAP is ideal for finding the best partitioning solution for small applications, while MCGAP is a sub-optimal solution for fast, large-scale applications.

IV. COMPUTATION OFFLOADING

Computation offloading is regarded as an effective way to guarantee user service quality by offloading the compute-intensive or latency-sensitive tasks to the edge devices or nearby edge servers [72].The main purpose of computation offloading is to reduce the response delay of the service and improve the service quality. In addition, when the edge node does not have the processing capability, the computation can be migrated to the edge server or the cloud data center to improve the overall performance of the system. In order to make the computation offloading decision, various aspects must be considered, such as performance maximization and energy consumption minimization. There are several questions that must be answered before computation offloading, such as:

(1) Can the task be offloaded? The task scheduler must determine if the task can be offloaded, i.e., what to offload, partial or total offload?

(2) When to offload the task? The task scheduler must determine the time slot for offloading under different constraints.

(3) Where to offload? The question is translated as which location is the best for offloaded workload execution, according to available resources distribution.

(4) Which offload policy will be adopted? That is, what's the main objective of the workload offloading, single performance metric maximization, or joint optimization and tradeoffing among multiple objectives? For example, massive edge devices are heterogeneous in terms of architecture, performance metrics, and power supply modes, which results in highly heterogeneous energy efficiency distribution among devices. Moreover, dynamic changes of network bandwidth and latency between cloud data center and edge equipment may lead to changes in energy consumption of data transmission. Therefore, different computation offloading policy leads to different power consumption. Therefore, a good computation offloading policy must find the optimal balance between the overall computation delay, data transmission, and related performance metrics.

In this section, we will survey the existing work on computation offloading and identify some challenges and future research directions.

A. WHAT TO OFFLAOD: THE SELECTION OF OFFLOADED WORKLOAD

The edge computing's promise for reducing service latency and network bandwidth usage can't realize unless the workload in cloud data centers can offloaded to edge devices and edge servers. Therefore, in the edge-cloud co-existing environment, original workloads executed at cloud data centers must be partitioned and some of them must be selected to run on edge devices and edge servers. Moreover, in some cases like IoT, local processing power is heterogeneously distributed across the large number of heterogeneous devices, and the local computing resources are not enough to run complex applications. Thus careful selection of offloaded workload to edge devices can help achieve lower latency and system performance.

Caching for content data locally is the ordinary solution for faster content delivery in many applications. Similarly, caching data from the cloud data centers to local edge devices or nearby edge servers can also provide lower latency for content delivery. For example, nowadays' web server contains many dynamically generated web pages, and the dynamic pages dominate network traffic, especially dynamic contents like music or video streaming.

To improve quality of experiences (QoE), caching content data close to end users, offloading data processing to the proxy servers, or caching fragments of dynamic pages and performing page composition after user's page access, are suitable strategies. Yuan *et al.* [73] proposed not to migrate the centralized database to the client, and offloading and caching on the edge server to reduce application latency. In addition, filtering a large number of server requests into the web proxies can significantly reduce server side workload. Chen *et al.* [74] proposed the network caching mechanism to utilize the storage capacity of diverse network devices to save network traffic.

To alleviate the pressure of rapid growth in demand for caching and computing services, Zhou *et al.* [75] proposed a new information centric heterogeneous network framework for content caching and computing. They investigated the problem of virtual resources allocation for communication, computing, and caching and the allocation problem is formulated as a joint optimization problem under constraints of caching, computation, and system virtualization. To solve this joint problem, they design a distributed algorithm based on the multiplier alternating direction method (MADM) which allows each infrastructure provider to solve its own problems without exchanging channel state information to reduce computation complexity and traffic overhead.

Lin *et al.* [76] proposed to update data intensive edge computing applications in the core database, which allows the application to be adjusted without sending a copy of the database to the edge. They also propose a wide-area replication protocol to provide dynamic content delivery while leveraging the advantages of edge computing.

Except the content data caching, coarse grained workload offloading, such as computation and analysis can also be offloaded, which consist of multiple primitive data processing operations in edge devices. In energy-constrained mobile platform, computation intensive applications are prone for performance downgrading. To extend battery life of mobile system, Kumar and Lu [77] proposed to perform the computation elsewhere rather than the mobile system itself to save energy. Specifically, the results show that cloud computing can save energy for mobile users, but not all applications are energy efficient. Moreover, mobile cloud computing must provide energy savings, as well as data reliability, privacy, and energy consumption.

Traditional computing offloading requires the transfer of user-entered data from the edge device to the cloud data center or edge server prior to computation, which is known as offline prefetching. However, the offline prefetching may cause heavy network communication traffic. Ko *et al.* [78] proposed a real-time data prefetching architecture for mobile computing offloading based on task-level computing prefetching and cloud computing simultaneously. The proposed approach controls the size of the corresponding prefetching data to minimize the energy consumption by dynamically selecting the prefetching task, avoiding excessive data offloading but retaining the advantage of reducing application execution time and power consumption through workload prediction. They also proposed an optimal and suboptimal data prefetching strategy to enable mobile devices to prefetching offloaded data within a given energy and time limits by using complex predictions of subsequent states of the communication channel.

B. WHEN TO OFFLOAD: THE PRECISE TIMING

Computation offloading can leverage the capabilities of computing, storage, networking and energy of edge devices, and provide lower latency for computation-intensive applications and services. However, since the network conditions are dynamically changing during application execution, decision on workload offloading must determine when the workload should be offloaded. In other words, the task scheduler must precisely time the offload opportunity considering all conditions and system status. For example, data caching during network congestion may improve significantly the system performance, while transferring large volume data to cloud data center is possible given that the link to cloud data center is enough for data communication.

In the previous section, we explored the problem of selection of offloaded workload, including data caching, data storage and computation and analysis offloading. In this section, we will discuss the work towards the problem of when to offload. The question of *when to offload* can be translated into the question that at what exact timing slots that the workload offloading can achieve the best performance gains and minimal costs or overheads, including energy consumption and bandwidth usage. Once upon the computation offloading is decided, the data and task will be partitioned into fractions. Due to dynamics of network connection and edge devices' availability, precise timing for workload offloading is the key to provide better system performance and less resource usage. Moreover, execution order of partitioned workload can also have impact on the system performance. Therefore, system monitoring and workload characterization including task arrival rates and deadlines can help make better offloading decision.

For example, modern processor are equipped with capability of dynamic voltage and frequency scaling(DVFS) to change operating voltage and frequency to save power and energy according to workload intensity. Therefore, running applications with different operating voltage and frequency may have significant impact on application performance. To leverage the DVFS capability, Wang *et al.* [79] proposed to use DVFS in the computation offloading on smart mobile devices, which enables smart mobile devices to adjust computational speeds based on computation demands dynamically to reduce energy consumption and computation time. Specifically, they optimize the processor speed, transmission power, and offload rate on smart mobile devices to minimize energy consumption and application execution delays. Since the existing computing offload strategies are not directly applicable to smart mobile devices that use DVFS techniques, they proposed a new computation offloading strategy for single-server and multi-server scenarios.

C. WHERE TO OFFLOAD: THE SCHEDULING OF OFFLOADED WORKLOADS

The workload offloading can be finally implemented via scheduling partitioned tasks to targeted edge devices and edge servers. The selection of targeted edge devices and edge servers involves the multiple objective optimization including performance, energy, network bandwidth, and data privacy protection methodology. For example, an intrinsic scheduling policy is energy hungry tasks are offloaded to the cloud servers to save energy, while data intensive tasks are offloaded to the edge servers to provide lower latency and less network traffic.

More specifically, offloaded task scheduling should consider the whole system status, including network status, task requirements, device information, etc. For an instance, if the network bandwidth is sufficient, cloud servers can be chosen for workload execution, otherwise edge servers or local devices are target places for workload execution. Moreover, if the task requires low latency, the edge servers are the perfect location for task execution.*MpOS* [80] is one of the offloading frameworks that perform all operations related to the offloading decision on the mobile device based on decision tree. The *MpOS* framework can reduce the energy consumption of mobile devices by using the proposed adaptive monitoring method and can reduce the power consumption up to 55%.

Effective offloading decisions are made after the inference of where offloading will improve system performance or get maximal gains. The problem is that making such an offloading decision relies on monitoring several parameters periodically, but these monitoring usually are computation intensive tasks that can cause additional overheads when running on a mobile device. To this end, Rego *et al.* [89] proposed a new method for defining an offloading strategy using a decision tree. In their method, all computation intensive operations related to the offloading decision creation are transferred to the remote server for execution, while the mobile device only needs to parse the constructed decision tree previously. They also proposed an adaptive monitoring scheme that is unique in that the mobile cloud computing system monitors the metrics associated with the offload decision under this scenario only and uses this information to make on-demand changes to the list of metric parameters monitored by the system.

Mobile devices need to measure a number of factors including resources, latency, security, and privacy when choosing an edge node or cloud server for workload offload. Meurisch *et al.* [81] proposed an approach to make offloading decisions with knowledge of the disadvantages of the current service running status on the offloading system in advance for all current computation offloading methods. Firstly, they detect and query unknown available targeted offloading destinations, such as nearby edge nodes, cloudlets, or remote clouds, in an energy efficient manner at runtime to make better offloading decisions. The principle is to evaluate the unknown offloading system by offloading the micro tasks and use the regression model to predict the performance and cost of offloading larger tasks. Then, the decision support for computational offloading are conducted. The experimental results show that the proposed approach can predict the performance of the unknown offloading platform with an accuracy of 85.5%.

D. ENERGY AND QOS TRADEOFF BETWEEN COMPUTATION AND DATA COMMUNICATION

Currently, more and more applications are running on smart mobile devices, and the user's quality of experience is the most important indicator to measure the success of applications and devices. However, such smart mobile devices at the edge of the network usually have limited resources including computing power, storage space and battery capacity, which makes it difficult to meet the growing needs of mobile users. In order to provide higher quality services, resources must be allocated and scheduled according to user requirements and services level agreements (SLAs). Therefore, delay-sensitive applications must be prioritized, and computation intensive applications should get enough computing resources. To this end, quality of experience (QoE) represents a user's subjective perception of the quality of service and performance of devices, networks, systems and applications. Computation offloading to the edge servers and then returning the computation results to the mobile devices can significantly alleviate the resource demand of the smart mobile device. In the process of computation offloading, it is necessary to introduce and implement the requirements of QoS and QoE, formulate a reasonable task offloading sequence, and determine the offloading timing of each task.

Moreover, the increasing number of edge nodes and the pervasive data analysis services on them causes the energy consumption of both the edge and data centers to increase significantly. Moreover, high energy consumption can lead to higher system operating cost and lower system reliability. Therefore, energy awareness is also the key factor that can help make more optimal computation offloading decision [82]–[85]. Saving energy and enhancing processing power are the benefits of computing offloading to mobile devices, but communication between mobile devices and edge nodes and cloud servers can cause certain execution delays, which can affect application's performance. Therefore, the balance between computing and communication is critical for computation offloading. Wang *et al.* [86] proposed a joint optimization problem that uses computation offloading to reduce the energy consumption of mobile devices while minimizing application execution latency. They formulated the problem as *MinED* and a 0-1 integer linear programming (ILP) problem and gave the optimal solution during polynomial time based on weighted double matching for special cases where there is sufficient residual energy on the mobile device and each application requires the same amount of resources.

Deng *et al.* [87] argued that with the rapid development of mobile applications and the increase of computational complexity, user mobility and fault tolerance should also be considered in the design and implementation of computation offloading strategies in mobile cloud computing and service execution process. They proposed a new computation offloading strategy by using genetic algorithm optimization. Similarly, Wang and Li [71] proposed a computation offloading approach in mobile cloud computing system and a context-aware offloading algorithm. They presented a general cost estimation model for cloud resources to estimate task execution costs, including execution time and energy consumption. Wang's offloading algorithm can provide offloading decisions and locations at runtime by referring to the context changes of network status, device information and the availability of multiple cloud resources. The proposed offloading approach considers a variety of cloud resources, such as mobile ad-hoc networks, cloudlets, and public clouds, to provide adaptive mobile cloud computing services using the proposed offloading algorithm and cost models. However, the proposed algorithm doesn't consider the communication between different cloud resources, where network communication may have significant impact on the performance of the prototype system in terms of device fault tolerance.

To ensure that the edge node storage system efficiently stores and accesses continuous uninterrupted real-time data, it has been proposed to deploy high-density, low-power, low-latency, and high-write nonvolatile storage media like nonvolatile memory (NVMe) at the edge device. In order to realize the edge programmability of the edge computing and reduce the power consumption of the code at runtime, power profiling and accounting support to the system call and runtime library can help the edge computing operating system to perform power consumption decomposition and prediction on the source code level accurately.

V. CASE STUDIES OF COMPUTATION OFFLOADING

The limitation of available remote resources will affect the scalability of computation offloading. Most of the current researches on computation offloading only focus on the computing offloading to a single server. However, multisite computation offloading is also promising and feasible for more energy savings. In this section, we introduce some existing computation offloading methods in, which are divided into two main categories, i.e., the gaming and cooperation

TABLE 3. Some existing work on computation offloading.

Work	Contributions
Dinh *et al.* [8]	framework of computation offloading from single mobile device (MD) to multiple edge devices
Cao *et al.* [10]	MEC Joint computing and collaborative communication
Lyu *et al.* [11]	New integrated architecture for cloud, MEC and IoT
Guo *et al.* [12]	Cloud - MEC collaborative computing uninstallation problem
Rimal *et al.* [13]	WiFi access network providing MEC function
Deng *et al.* [14]	Multi-cell mobile edge computing
Zheng *et al.* [17]	Joint downlink and uplink, Ultra-Dense HetNets
Alam *et al.* [20]	Autonomic computation offloading
Qiu *et al.* [21]	Blockchain empowered mobile edge computing
Yu *et al.* [25]	deep learning approach
Xu *et al.* [27]	Online learning for offloading
Guo *et al.* [28]	Ultradense IoT networks
Liu *et al.* [32]	Ultra-reliable low latency communications
Guo *et al.* [34]	Fiber-wireless networks
Ning *et al.* [90]	cooperative partial computation offloading scheme
Wang *et al.* [60]	ENORMc, resource management
Wang *et al.* [71]	Computation offloading scheme on handheld devices
Mach *et al.* [73]	Review
Chen *et al.* [75]	5G communication
Rego *et al.* [91]	Decision tree
Zhang *et al.* [92]	Cloud-based VEC offloading framework
Liu *et al.* [93]	Stackelberg game
Meskar *et al.* [94]	Game, Gauss-Seidel Competitive game
Chen *et al.* [95]	Game, Distributed computing offloading
Jia *et al.* [96]	Online offloading
Wang *et al.* [71]	Context awareness
Lin *et al.* [97]	Triple decision maker(TDM)
Zhou *et al.* [98]	Code offloading framework (mCloud)
Kuang *et al.* [99]	Agent based
Kao *et al.* [100]	Hermes, Mobile devices
Deng *et al.* [88] Sardellitti *et al.* [101] Terefe *et al* [102]	Mobile devices

between the edge and the cloud, and the heuristics based offloading. We list some existing work on computation offloading in Table 3.

A. GAMING AND COOPERATION BETWEEN EDGE AND THE CLOUD

When considering the computation offloading in edge computing, it is necessary to consider the gaming and cooperation between the edge and the cloud for task scheduling and collaboration. For example, if there are multiple edge devices which are suitable candidates for computation offloading in a distributed edge computing environment across multiple administrative domains, different selection of the targeted edge devices may result in different system performance and gains. More specifically, if the candidate edge devices contend to run offloaded workload on themselves, they may promise different resources and service provisioning.

For example, in connected autonomous vehicle systems, although the service can be improved by computation offloading, the mobile edge computing server may become the performance bottleneck during heavy vehicle traffic. To solve this problem, Zhang *et al.* [90] proposed a layered cloud-based vehicle edge computing (VEC) offloading framework that compensates for the shortcomings of the MEC server's computation resources by sharing backup servers nearby. They designed a multi-layer optimal offloading scheme using Stackelberg gaming theory where the incentive mechanism is introduced into the selection of the offloading server and the allocation of computing resources. The VEC server allocates computing resources to the vehicle through incentives that maximize the utility of the vehicle and computing server dynamically. They also proposed a distributed algorithm that can increase the service provider's revenue while still meeting the task delay requirements in order to obtain an optimal offloading strategy.

For the computation offloading scenarios of multiple mobile users, Sardellitti *et al.* [99] investigated MIMO(multi-input multi-output) multi-cell systems where multiple mobile users require that the computing task can be offloaded to the cloud data centers. They modeled the offloading problem as the joint optimization of radio resources to minimize overall energy consumption while meeting delay constraints. In the case of single user, they proposed approach can find the global optimal solution in closed form. However, in multi-user scenarios, an iterative algorithm based on convex optimization is proposed to computing the local optimal solution. The main idea is to compute the optimal resources allocation by exchanging information with the wireless access base station.

Ning *et al.* [88] proposed a cooperative partial computation offloading scheme for mobile edge computing enabled IoT. However, there is a potential for local edge servers that they may not be willing to participate in computation offloading, which requires more incentives to stimulate cloud service operators and edge server owners to participate in computation offloading. Specifically, when performing computation offloading, the scheduler need to know how much computations each local edge server can provide and how much payment the edge server owner ask for. Liu *et al.* [91] modeled the economic interaction between cloud service operators and edge server owners as a Stackelberg game, which enables cloud service operators to allocate computation based on the valuation of edge servers to maximize the benefits of cloud service operators and edge servers. In real world scenario, an edge server owner can participate or leave a computation offloading arbitrarily. They separate the Stackelberg game into two phases:

(1) The first phase: the cloud service operator provides a payment profile to motivate the edge server owner to participate in the computation offloading;

(2) The second phase, the edge server owner replies to the payment data based on the amount of computation that can provide and gives the best payment strategy for cloud service operators and computing offload strategy for edge server. They analyze the equilibrium existing in the Stackelberg game and prove that there is *Nash* equilibrium in the game between the cloud service operator and the edge server owner. Furthermore, they design two computational offloading algorithms, a low-latency algorithm for a single-round Stackelberg game and a low-complexity algorithm for multiple rounds of Stackelberg games.

However, under the scenario that the local edge server may participate or withdraw from the computation offloading, only the computation amount of the edge server and the computation allocation of cloud server operator's is given, but it is not proved whether the edge server joins and exits the computing task arbitrarily has an impact on the computation and whether the reward mechanism can guarantee user's QoS and QoE.

Regarding computation offloading of competing users on shared channels, Meskar *et al.* [92] investigated a group of mobile users using cloud computing offloading who offload computing tasks to the cloud server through shared transport channels to reduce energy consumption on the shared channel. In their model, time slices are distributed in a round-robin fashion to mobile users who need computation offloading. Therefore, offloading tasks have strict constraints on execution time. They model the system as a competitive game in which each user attempts to contend for a shared channel to reduce their energy consumption. The game is proven to *Nash* Equilibrium and subject to real-time constraints on task execution time, user channel bit rate, and shared channel contention. Specifically, each user can make an offloading decision based on the information obtained from the central cloud controller independently, while the central controller cannot apply the offloading decision to the user directly but can modify the information obtained by the user to influence the formulation of the offloading decision.

Moreover, in an edge computing environment with multiple users contending for shared resources such as network bandwidth, resource contention will cause serious interference and reduce the speed of data transmission. Chen [93] argued that gaming theory is a suitable for making decentralized and self-organized computation offloading decision, and the self-organizing function can increase the maintainability of the edge computing system and alleviate the heavy burden of centralized management of the cloud automatically. Similarly, Chen *et al.* [101] proposed gaming theory approach to make computation offloading decision of multiple mobile device users in mobile edge cloud computing in multi-channel wireless interference environment.

B. HEURISTICS BASED COMPUTATION OFFLOADING

Heuristics based offloading is a common offloading method in edge computing and it typically contains two parts:

(1) Searching. The searching phase is to search for a set of best tasks as the initial computation offloading group according to the offloading requirement;

(2) Adjustment. The adjustment phase is to adjusting the offloading strategy according to the network bandwidth, the number of available edge cloud servers, and the maximum utility ranking result of the system during the computation offloading process.

However, online heuristics based computation offloading needs elaborate modeling and design of the heuristics algorithm. Jia *et al.* [94] proposed an online task offloading algorithm to minimize the execution time of applications on mobile devices. They found that for parallel tasks, load balancing heuristics can be used to offload tasks into the cloud to maximize parallelism between mobile and cloud. The rationale of the proposed algorithm is that if a task will be offloaded, its adjacent tasks may also be offloaded. Since offloading tasks to the cloud data centers can reduce the task's execution time, maximizing parallelism of execution of offloaded task between clouds and mobile devices can provide lower latency.

Different from heuristic algorithms that don't provide theoretical performance guarantees, Kao *et al.* [98] proposed *Hermes*, a fully polynomial time approximation scheme, to tradeoff delays and resources consumption within acceptable performance constraints. For task assignments that balance latency and resource consumption, *Hermes* suggests strategies that outperform greedy heuristic algorithms and reduced latency up to 16% compared with the heuristic algorithms.

Kuang *et al.* [97] proposed an agent-based offloading framework for mobile cloud computing to decrease the request delay of mobile users, which can alleviate the overheads of network communication and reduce the excess energy consumption caused by invalid transmission requests. They formulate the problem of maximizing user's energy saving under task execution time and bandwidth constraints, and designed a dynamic programming after filtering (DPAF) algorithm. And they transform the original offloading problem to the classic 0-1 Knapsack problem by the filtering process on the agent, and adopt dynamic programming algorithm to find an optimal offloading strategy.

Nowadays smartphones are popular platforms to execute offloaded computation tasks although there are delays due to the uploading and downloading of tasks. Lin *et al.* [95] proposed an offloading framework, namely, Triple Decision Makers (TDM) to reduce power consumption and response time. They presented a customizable cost function for the conflicting goals of response time and energy consumption, and a lightweight analysis method to evaluate performance and energy consumption during the offloading process. The cost function takes into account the factors such as bandwidth, CPU speed, memory bandwidth, etc.

Offloading tasks to a remote or edge cloud can help mobile devices' save energy and gain more computing power, but it also brings additional transmission energy consumption and latency. Therefore, a good offloading strategy should implement partial offloading tasks and be able to weigh the relationship between offloading advantages and additional costs. Zhou *et al.* [96] proposed a context-aware offloading framework, *mCloud*. The *mCloud* consists of mobile devices, cloudlets and public cloud services, which provide adaptive mobile offloading service to improve the performance and availability of mobile cloud computing services in terms of better performance and less energy consumption.

Terefe *et al.* [100] proposed a multi-site offloading strategy for mobile devices. By analyzing the data-intensive and computation-intensive modules of the application, they use a mathematical model to simulate the energy consumption of the multi-site application execution. In order to better adapt to the changing network bandwidth, they use finite state discrete time *Markov* chain to simulate the communication channel based offloading, and model the multi-sites offloading decision problem as the shortest path problem on a directed acyclic graph (DAG) through Markov decision framework. They designed an energy-efficient multi-site offloading algorithm to determine the optimal offloading decision for multi-site program execution through multiple iterations. However, it would be better to extend the data-intensive and computation-intensive oriented modeling implementation and verification in real edge computing environment.

Mao *et al.* [102] investigated the mobile edge cloud system with energy harvesting equipment, and design a dynamic computation offloading strategy. They used the cost of execution delay and task failure as a performance metric to evaluate the offloading strategy, and proposed a dynamic computation offloading algorithm based on *Lyapunov* optimization. Moreover, they use DVFS to optimize the computation offloading and data transmission process. The results show that the algorithm can achieve the optimal performance by adjusting the DVFS parameters and can not only reduce the execution cost effectively, but also reduce the task failure successfully.

Zhang *et al.* [101] investigated the design of computation offloading mechanism of mobile edge computing in 5G heterogeneous networks and proposed a multi-device energy-saving computation offloading framework, the energy-efficient computation offloading (EECO), to reduce the energy consumption of computing tasks during computation offloading and execution. They tried to reduce the energy consumption of the system while satisfying the delay constraints and classify and assign appropriate priority to mobile devices. They solved the optimization problem under polynomial complexity by the three-stage energy saving computation offload scheme.

VI. PERFORMANCE EVALUATION AND SIMULATION

Offloading computation on the mobile edge cloud can avoid large scale data movement to achieve fast response, controllable service delay, low energy consumption and other performance characteristics than cloud computing. Moreover, elaborate performance evaluation of current edge computing platforms and systems can provide insights and suggestions to improve the systems in terms of latency, throughput, energy consumption, and privacy preservation. However, most of the existing edge computing research works focus on minimizing latency and energy consumption, while improving and optimization of a single performance index often sacrifices other performance metrics. Therefore, proper performance characteristics and metrics are lacking in evaluating current edge computing systems. For example, since the edge network near the user is often connected with a variety of edge devices, which makes the edge computing environment more complex and changeable in terms of performance characteristics than the centralized clouds. Performance indicators, such as energy consumption, network bandwidth, computing speed, data security, response delay, and privacy protection, must be considered to evaluate and understand the various performance characteristics of edge computing and its computation offloading. \

Tao *et al.* [44] investigated how to make use of the performance characteristics of network edge to achieve reasonable flow splitting scheduling of computing tasks on the premise of ensuring the performance of mobile edge computing services. They formulated the resource, time-delay limited service quality assurance and energy saving of mobile devices as a joint optimization problem, and designed a flow-splitting algorithm to make computation offloading decisions for each mobile device by considering energy-consumption and task completion time.

General and universal evaluation of edge computing systems can also provide controllable and repeatable experiments. However, it's very expensive to construct a real edge computing testbed to in terms of hardware deployment, application development, user interaction, and network traffic generation. For example, even for IoT [104]–[106] tested, it can produce massive data which is difficult to deal with for cloud computing. To this end, Harshit *et al.* [107] proposed a simulator, called *iFogSim*, to model IoT and Fog environments and measure the impact of resource management techniques in latency, network congestion, energy consumption, and cost. However, *iFogSim*could be extended in multiple dimensions in terms of energy consumption and data privacy preservation.

To evaluate the various performance aspects of edge computing including computation offloading and resource allocation, performance evaluation tools are critical to both researchers and engineers to design resource management techniques. However, the research community is lack of such performance tools or simulator to help researcher to design better resource allocation and computation offloading algorithms under various constraints.

VII. CONCLUSION

Edge computing is emerging as one of the strategic technology that will redefine the future computing paradigm for its promise of lower latency, less bandwidth usage and data privacy protection. Computation offloading is critical to make the promise into reality in various application scenarios, from connected autonomous vehicles to smart home. Decision on computation offloading involves sophisticated resource management and allocation among multiple parties in the cloud–edge collaborative environment. In this paper a thorough research survey is conducted to reveal the state-of-the-art of computation offloading in edge computing. Various aspects of computation offloading, including energy consumption minimization, Quality of Services guarantee, and Quality of Experiences enhancement are surveyed. Moreover, resource allocation approaches, gaming theory and heuristics based computation offloading optimization of system performance and overheads for computation offloading decision making are also surveyed. Our work presented in this paper can help not only the research community but also industrial practitioner to understand the state-of-the-art of computation offloading in edge computing to design better systems with elaborate resource management and computing placement mechanism.

REFERENCES

[1] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, Feb. 2018.

[2] B. Zhou and R. Buyya, "Augmentation techniques for mobile cloud computing: A taxonomy, survey, and future directions," *ACM Comput. Surv.*, vol. 51, no. 1, Apr. 2018, Art. no. 13.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[4] W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, V. B. C. Souza, A. Jukan, G-J. Ren, and O. G. de Dios, "Evaluating the benefits of combined and continuous fog-to-cloud architectures," *Comput. Commun.*, vol. 113, pp. 43–52, Nov. 2017.

[5] X. Masip-Bruin, E. Marin-Tordera, A. Jukan, and G.-J. Ren, "Managing resources continuity from the edge to the cloud: Architecture and performance," *Future Gener. Comput. Syst.*, vol. 79, pp. 777–785, Feb. 2017.

[6] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[7] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[8] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[9] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[10] X. Cao, F. Wang, R. Zhang, S. Cui, and J. Xu, "Joint computation and communication cooperation for mobile edge computing," presented at the 16th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt), May 2018, pp. 1–6.

[11] X. Lyu, H. Tian, and L. Jiang, "Selective offloading in mobile edge computing for the green Internet of Things," *IEEE Netw.*, vol. 32, no. 1, pp. 54–60, Jan./Feb. 2018.

[12] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[13] B. P. Rimal, D. P. Van, and M. Maier, "Mobile edge computing empowered fiber-wireless access networks in the 5G era," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 192–200, Feb. 2017.

[14] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.

[15] M. Hu, L. Zhuang, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Learning driven computation offloading for asymmetrically informed edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1802–1815, Aug. 2019.

[16] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, and N. Georgalas, "Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 64–69, May 2019.

[17] J. Zheng, L. Gao, H. Wang, X. Li, P. Xu, L. Wang, B. Jiang, and X. Yang, "Joint downlink and uplink edge computing offloading in ultra-dense HetNets," *Mobile Netw. Appl.*, to be published. doi: 10.1007/s11036-019-01274-y.

[18] B. Huang, Z. Li, P. Tang, S. Wang, J. Zhao, H. Hu, W. Li, and V. Chang, "Security modeling and efficient computation offloading for service workflow in mobile edge computing," *Future Gener. Comput. Syst.*, vol. 97, pp. 755–774, Aug. 2019.

[19] D. Zhang, L. Tan, J. Ren, M. K. Awad, S. Zhang, Y. Zhang, and P-J. Wan, "Near-optimal and truthful online auction for computation offloading in green edge-computing systems," *IEEE Trans. Mobile Comput.*, to be published.

[20] M. G. R. Alam, M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for IoT applications," *Future Gener. Comput. Syst.*, vol. 90, pp. 149–157, Jan. 2019.

[21] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, Aug. 2019.

[22] L. Tang and S. He, "Multi-user computation offloading in mobile edge computing: A behavioral perspective," *IEEE Netw.*, vol. 32, no. 1, pp. 48–53, Jan./Feb. 2018.

[23] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[24] M. A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, "Computation offloading game for an UAV network in mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[25] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: A deep learning approach," *IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[26] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, "Energy efficient task caching and offloading for mobile edge computing," *IEEE Access*, vol. 6, pp. 11365–11373, 2018.

[27] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[28] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultradense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.

[29] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[30] N. T. Ti and L. B. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[31] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Proc. 8th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Sep. 2016, pp. 288–294.

[32] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[33] K. Sato and T. Fujii, "Radio environment aware computation offloading with multiple mobile edge computing servers," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Mar. 2017, pp. 1–5.

[34] H. Guo, J. Liu, and H. Qin, "Collaborative mobile edge computation offloading for IoT over fiber-wireless networks," *IEEE Netw.*, vol. 32, no. 1, pp. 66–71, Jan./Feb. 2018.

[35] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1619–1632, Aug. 2018.

[36] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Trans. Services Comput.*, to be published.

[37] A. Asheralieva and D. Niyato, "Hierarchical game-theoretic and reinforcement learning framework for computational offloading in UAV-enabled mobile edge computing networks with multiple service providers," *IEEE Internet Things J.*, to be published.

[38] K. Bierzynski, A. Escobar, and M. Eberl, "Cloud, fog and edge: Cooperation for the future?" presented at the 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC), May 2017, pp. 62–67.

[39] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani,, "Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.

[40] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, "MIFaaS: A mobile-IoT-federation-as-a-service model for dynamic cooperation of IoT cloud providers," *Future Generat. Comput. Syst.*, vol. 70, pp. 126–137, May 2017.

[41] N. Mäkitalo, A. Ometov, J. Kannisto, S. Andreev, Y. Koucheryavy, and T. Mikkonen, "Safe, secure executions at the network edge: Coordinating cloud, edge, and fog computing," *IEEE Softw.*, vol. 35, no. 1, pp. 30–37, Jan./Feb. 2018.

[42] S. Bi and Y. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[43] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[44] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 774–777, Dec. 2017.

[45] D. Sabella, M. Filippou, I. Karls, Y. Yang, J. Zhu, and K. Roth, "Application computation offloading for mobile edge computing," U.S. Patent 15 855 652, Dec. 27, 2017.

[46] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," presented at the IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2017, pp. 1–6.

[47] Q. Li, J. Zhao, Q. Zhang, and Y. Gong, "Energy-efficient computation offloading and resource allocation in fog computing for Internet of Everything," *China Commun.*, vol. 16, no. 3, pp. 32–41, Mar. 2019.

[48] C. You, Y. Zeng, K. Huang, and R. Zhang, "Asynchronous mobile-edge computation offloading: Energy-efficient resource management," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7590–7605, Nov. 2018.

[49] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[50] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[51] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, and J. Henkel, "Computation offloading and resource allocation for low-power IoT edge devices," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 7–12.

[52] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[53] W. S. Shi, F. Liu, H. Sun, and Q. Q. Pei, *Edge Computing*, 1st ed. Beijing, China: Science Press, 2018.

[54] W. S. Shi, H. Sun, J. Cao, Q. Zhang, and W. Liu, "Edge computing—An emerging computing model for the Internet of everything era," *J. Comput. Res. Develop.*, vol. 54, no. 5, pp. 907–924, 2017.

[55] Z. Li, X. Peng, L. Chao, and Z. Xu, "EveryLite: A lightweight scripting language for micro tasks in IoT systems," in *Proc. 3rd ACM/IEEE Symp. Edge Comput.*, Oct. 2018, pp. 381–386.

[56] Q. Zhang, X. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: Big data sharing and processing in collaborative edge environment," in *Proc. 4th IEEE Workshop Hot Topics Web Syst. Technol.*, Washington, DC, USA, Oct. 2016, pp. 20–25.

[57] J. Chabas, C. Gnanasambandam, S. Gupte, and M. Mahdavian, "New demand, new markets: What edge computing means for hardware companies," McKinsey & Company, New York, NY, USA, Tech. Rep., 2018. [Online]. Available: https://www.mckinsey.com/industries/high-tech/our-insights/new-demand-new-markets-what-edge-computing-means-for-hardware-companies?hlkid=9c60010d1b394427a06c47feb2e3401c&hctky=2618809&hdpid=1dbd559b-1690-4c8b-bc99-35b7ed7279a2

[58] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multi-objective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Jan. 2018.

[59] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "ENORM: A framework for edge node resource management," *IEEE Trans. Services Comput.*, to be published.

[60] Z. Tan, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Virtual resource allocation for heterogeneous services in full duplex-enabled SCNs with mobile edge computing and caching," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1794–1808, Feb. 2018.

[61] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[62] J. Xu and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[63] L. Yang, B. Liu, J. Cao, Y. Sahni, and Z. Y. Wang, "Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds," in *Proc. 10th Int. Conf. Cloud Comput.*, Honolulu, CA, USA, Jun. 2017, pp. 246–253.

[64] J. Liu, W. Song, and M. Atiquzzaman, "Bandwidth-adaptive partitioning for distributed execution optimization of mobile applications," *J. Netw. Comput. Appl.*, vol. 37, pp. 334–347, Jan. 2014.

[65] Y. Yin, L. Chen, Y. Xu, and J. Wan, "Location-aware service recommendation with enhanced probabilistic matrix factorization," *IEEE Access*, vol. 6, pp. 62815–62825, 2018.

[66] H. Gao, H. Huang, X. Yang, Y. Duan, and Y. Yin, "Toward service selection for workflow reconfiguration:An interface-based computing solution," *Future Gener. Comput. Syst.*, vol. 87, pp. 298–311, Oct. 2018.

[67] G. Xu, Y. Zhang, A. K. Sangaiah, X. Li, A. Castiglione, and X. Zheng, "CSP-E^2: An abuse-free contract signing protocol with low-storage TTP for energy-efficient electronic transaction ecosystems," *Inf. Sci.*, vol. 476, pp. 505–515, Feb. 2019.

[68] G. Q. Xu, J. Liu, Y. R. Lu, X. J. Zeng, Y. Zhang, and X. M. Li, "A novel efficient MAKA protocol with desynchronization for anonymous roaming service in global mobility Networks," *J. Netw. Comput. Appl.*, vol. 107, pp. 83–92, Apr. 2018.

[69] L. Qi, J. Yu, and Z. Zhou, "An invocation cost optimization method for Web services in cloud environment," *Sci. Program.*, vol. 2017, 2017, Art. no. 4358536.

[70] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, Apr. 2018, Art. no. 3075849.

[71] C. Wang and Z. Li, "A computation offloading scheme on handheld devices," *J. Parallel Distrib. Comput.*, vol. 64, no. 6, pp. 740–746, Jun. 2004.

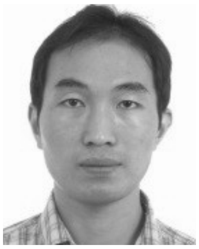
[72] P. Mach, Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[73] C. Yuan, Y. Chen, and Z. Zhang, "Evaluation of edge caching/off loading for dynamic content delivery," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1411–1423, Nov. 2004.

[74] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, "Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, p. 974, 2016.

[75] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11339–11351, Dec. 2017.

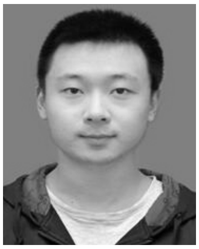
[76] Y. Lin, B. Kemme, M. Patino-Martinez, and R. Jimenez-Peris, "Enhancing edge computing with database replication," in *Proc. 26th IEEE Int. Symp. Reliable Distrib. Syst. (SRDS)*, Oct. 2007, pp. 45–54.
[77] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.
[78] S.-W. Ko, K. Huang, S.-L. Kim, and H. Chae, "Live prefetching for mobile computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3057–3071, May 2017.
[79] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.
[80] P. B. Costa, P. A. L. Rego, L. S. Rocha, F. A. M. Trinta, and J. N. de Souza, "MpOS: A multiplatform offloading system," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Salamanca, Spain, Apr. 2015, pp. 577–584.
[81] C. Meurisch, J. Gedeon, T. A. B. Nguyen, F. Kaup, and M. Muhlhauser, "Decision support for computational offloading by probing unknown services," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Vancouver, BC, Canada, Jul./Aug. 2017, pp. 1–9.
[82] C. Jiang, G. Han, J. Lin, G. Jia, W. Shi, and J. Wan, "Characteristics of co-allocated online services and batch jobs in Internet data centers: A case study from Alibaba cloud," *IEEE Access*, vol. 7, pp. 22495–22508, 2019.
[83] Y. Qiu, C. Jiang, D. Ou, Y. Li, J. Wan, and Y. Wang, "Energy aware virtual machine scheduling in data centers," *Energies*, vol. 12, no. 4, p. 646, Feb. 2019.
[84] C. T. Fan, Y. Qiu, Y. Wu, and H. Zhang, "Interdomain I/O optimization in virtualized sensor networks," *Sensors*, vol. 18, no. 12, p. 4395, Dec. 2018.
[85] C. Jiang, Y. Wang, D. Ou, Y. Li, J. Zhang, J. Wan, B. Luo, and W. Shi, "Energy efficiency comparison of hypervisors," *Sustain. Comput., Inform. Syst.*, vol. 22, Jun. 2019, pp. 311–321.
[86] X. Wang, J. Wang, X. Wang, and X. Chen, "Energy and delay tradeoff for application offloading in mobile cloud computing," *IEEE Syst. J.*, vol. 11, no. 2, pp. 858–867, Jun. 2017.
[87] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, pp. 3317–3329, Dec. 2015.
[88] Z. Ning, P. Dong, F. Xia, and X. Kong, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.
[89] P. A. L. Rego, E. Cheong, E. F. Coutinho, F. A. M. Trinta, J. N. de Souza, and M. Z. Hasan, "Decision tree-based approaches for handling offloading decisions and performing adaptive monitoring in MCC systems," in *Proc. 5th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng.*, San Francisco, CA, USA, Apr. 2017, pp. 74–81.
[90] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.
[91] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach," *Comput. Netw.*, vol. 129, pp. 399–409, Dec. 2017.
[92] E. Meskar, T. D. Todd, D. Zhao, and G. Karakostas, "Energy aware offloading for competing users on a shared communication channel," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 87–96, Jan. 2017.
[93] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[94] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, Apr./May 2014, pp. 352–357.
[95] Y.-D. Lin, E. T.-H. Chu, Y.-C. Lai, and T.-J. Huang, "Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds," *IEEE Syst. J.*, vol. 9, no. 2, pp. 393–405, Jun. 2015.
[96] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "mCloud: A context-aware offloading framework for heterogeneous mobile cloud," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 797–810, Sep./Oct. 2017.
[97] Z. Kuang, S. Guo, J. Liu, and Y. Yang, "A quick-response framework for multi-user computation offloading in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 81, Apr. 2018, pp. 166–176.
[98] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Kowloon, Hong Kong, Apr./May 2015, pp. 1894–1902.
[99] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.
[100] M. B. Terefe, H. Lee, N. Heo, G. C. Fox, and S. Oh, "Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing," *Pervas. Mobile Comput.*, vol. 27, pp. 75–89, Apr. 2016.
[101] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[102] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[103] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
[104] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: An efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1506–1519, Apr. 2019.
[105] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. L. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based Cyber-physical systems," *World Wide Web J.*, to be published. doi: 10.1007/s11280-019-00684-y.
[106] L. Qi, P. Dai, J. Yu, Z. Zhou, and Y. Xu, "'Time–Location–Frequency'–aware Internet of things service selection based on historical records," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 1, pp. 1–9, 2017.
[107] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, 2017.



CONGFENG JIANG received the B.E. degree in hydro-electrical engineering from the North China University of Water Resources and Electric Power, in 2002, and the Ph.D. degree in hydro-electrical engineering from the Huazhong University of Science and Technology, in 2007. Since 2007, he has been an Assistant Professor with the School of Compute Science and Technology, Hangzhou Dianzi University, China. He is currently an Associate Professor with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou. He has published more than 50 articles in grid computing, cloud computing, virtualization, and big data systems. His research interests include system optimization and performance evaluation, and distributed system benchmarking. He is a member of ACM and CCF.

XIAOLAN CHENG is currently pursuing the M.S. degree with the School of Computer Science and Technology, Hangzhou Dianzi University, China. Her research interest includes edge computing.

HONGHAO GAO received the Ph.D. degree in computer science from Shanghai University, in 2012, where he started his academic career. His research interests include service computing, model checking-based software verification, and sensors data application. He is an IET Fellow, a BCS Fellow, an EAI Fellow, the CCF Senior Member, and the CAAI Senior Member.



XIN ZHOU is currently pursuing the M.S. degree with the School of Computer Science and Technology, Hangzhou Dianzi University, China. His research interest includes edge computing.

JIAN WAN received the B.S. degree in mechanical engineering, the M.S. degree in mathematics, and the Ph.D. degree in computer science from Zhejiang University, in 1990, 1993, and 1996, respectively. From 2000 to 2015, he was the Dean of the School of Compute Science and Technology, Hangzhou Dianzi University. He is currently the Vice President of the Zhejiang University of Science and Technology and also the Vice Director of the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou. He is also a Professor with the School of Information Engineering, Zhejiang University of Science and Technology, Hangzhou, China. He has published more than 150 articles in grid and services computing, cloud computing, virtualization, and distributed systems. His research interests include distributed systems, computer networks, and big data analytics.