Received: 6 June 2023 | *Revised: 25 October 2023* | *Accepted: 22 December 2023*
DOI: 10.37917/ijeee.21.1.9 *Vol. 21 | Issue 1 | June 2025*

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Review Article

IJEEE University of Basrah College of Engineering

A Review of Algorithms and Platforms for Offloading Decisions in Mobile Cloud Computing

Fatima Haitham Murtadha*, Suhad Faisal Behadili
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Correspondance
*Fatima Haitham Murtadha
Department of Computer Science, College of Science,
University of Baghdad, Baghdad, Iraq
Email: fatima.qetas2101m@sc.uobaghdad.edu.iq

Abstract
With the substantial growth of mobile applications and the emergence of cloud computing concepts, therefore mobile Cloud Computing (MCC) has been introduced as a potential mobile service technology. Mobile has limited resources, battery life, network bandwidth, storage, and processor, avoid mobile limitations by sending heavy computation to the cloud to get better performance in a short time, the operation of sending data, and get the result of computation call offloading. In this paper, a survey about offloading types is discussed that takes care of many issues such as offloading algorithms, platforms, metrics (that are used with this algorithm and its equations), mobile cloud architecture, and the advantages of using the mobile cloud. The trade-off between local execution of tasks on end-devices and remote execution on the cloud server for minimizing delay time and energy saving. In the form of a multi-objective optimization problem with a focus on reducing overall system power consumption and task execution latency, meta-heuristic algorithms are required to solve this problem which is considered as NP-hardness when the number of tasks is high. To get minimum cost (time and energy) apply partial offloading on specific jobs containing a number of tasks represented in sequences of zeros and ones for example (100111010), when each bit represents a task. The zeros mean the task will be executed in the cloud and the ones mean the task will be executed locally. The decision of processing tasks locally or remotely is important to balance resource utilization. The calculation of task completion time and energy consumption for each task determines which task from the whole job will be executed remotely (been offloaded) and which task will be executed locally. Calculate the total cost (time and energy) for the whole job and determine the minimum total cost. An optimization method based on metaheuristic methods is required to find the best solution. The genetic algorithm is suggested as a metaheuristic Algorithm for future work.

Keywords
MCC, Mobile Cloud, Offloading, MAUI, MEC.

I. Introduction

In recent years, increasing in use cloud, the cloud provides on-demand computing services over the Internet, and improves performance by providing powerful computing resources [1], the mobile faces a lot of limitations, especially with applications that require a lot of computation for example augmented reality, artificial intelligence, artificial vision, object tracking, image processing and natural language processing are becoming popular. Mobile Cloud Computing (MCC) is a hybrid of the three foundations of Cloud Computing, Mobile Computing, and Networking. Type of offloading regarding the access to network wired or wireless, also regarding when to take decisions either static or dynamic. However, the types of applications [2]. Such as multimedia (image processing), games, calculators, or predictors (AI apps). Also, the offloading types are according to the based-on algorithm. In general, offloading decisions are usually made by analyzing parameters including bandwidths, server speeds, available memory, server loads,

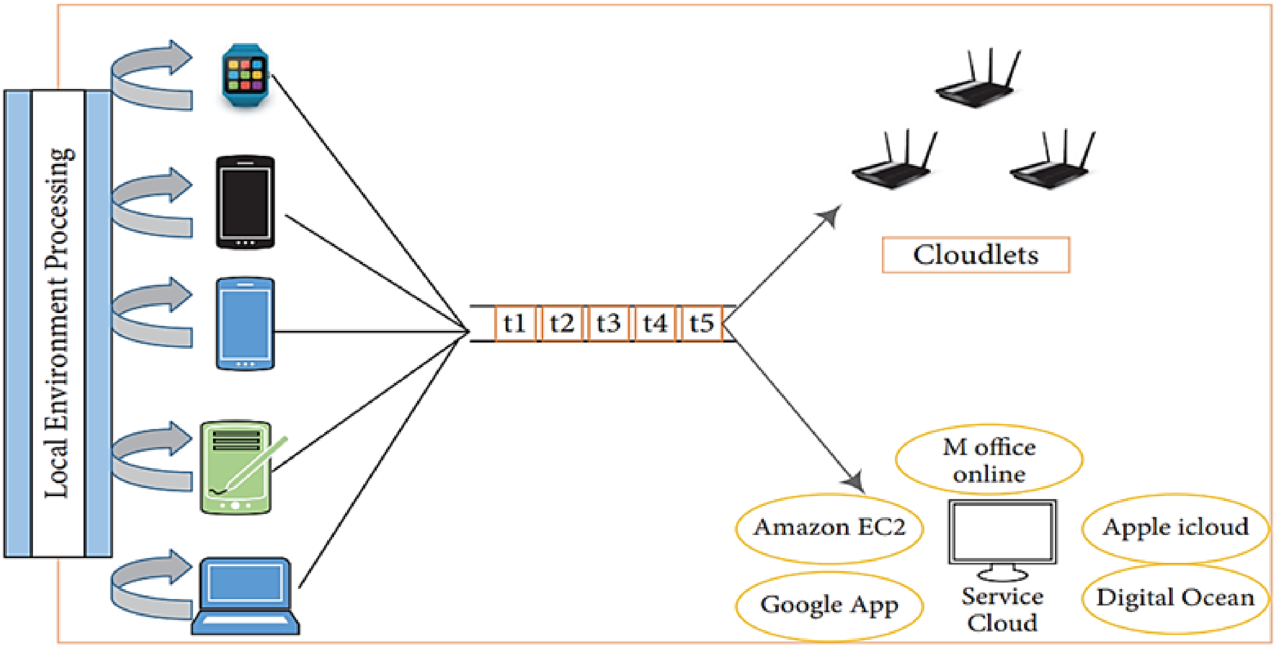

Fig. 1. Computational offloading in cloud [3].

and data amounts exchanged between servers and mobile systems to improve performance or save energy. Mobile has a lot of limitations such as CPU, memory, and battery life, to avoid these limitations the cloud has been utilized. Computation offloading mechanisms are the best solution so far, prompting the ecosystem to offload intensive computing functions to remote computing resources, such as edge-based servers shown in Fig. 1, which have huge computing resources for faster execution operations than local ecosystem resources [3].

II. Related Work

The MCC contributed several platforms and algorithms shown in Table I, hence the common was in the following:

A. ThinkAir

It provides a novel execution offload infrastructure and extensive resource consumption profiler for efficient and effective code migration; moreover, it provides library and compiler support [4], enabling developers to do minimal work on existing code The framework can easily used with modifications and also provides a VM manager in the cloud and a parallel processing engine to automatically manage smartphone VMs and task splitting and distribution across multiple virtual machines. It analyzes network health, program status, and device power to make offload decisions that reduce power consumption. However, the above framework only considers energy consumption as the only decision parameter.

B. ULOOF

It's an abbreviation of the term User Level Online Offloading Framework [5]. This platform implements a heuristic to develop this scheduler with decision Algorithms that minimize computational latency or power consumption. Decide Engines use a cost function to estimate the energy and time required for computation. They use empirical data to make such estimates. Provide a choice of how much power and latency developers want to prioritize. To estimate electricity consumption, they used a series of empirical studies using a different number of CPU cores and different clock speeds to create Energy model curves that map specific device power consumption.

C. MAUI

Maui. MAUI is a fine-grained approach that can remotely offload parts of programs to solve mobile energy problems [6]. The remote server can be a CC server or an EC server on a nearby Wi-Fi access point. As the pioneer of all offloading systems, MAUI offloading strategy takes advantage of program partitioning and full-process migration, which also reduces developers' programming workload. MAUI's architecture follows the client-server model. Both server and mobile devices have three functional components: agent, analyzer, and solver. Agents are used to transmit data and control instructions. The

TABLE I.
PLATFORMS AND ALGORITHM

Algorithm or Method	2025	Ref.	Implemented on	Number Servers	Working on	Metric to enhance
ThinkAir	2012	[4]	N-queens puzzle application face detection on 100 pictures	Multi-Servers	the ability of on-demand VM resource scaling and exploiting parallelism	execution time and energy
ULOOF	2018	[5]	mobile applications	Single and Multi-Servers	decrease energy consumption of mobile devices and the execution time of mobile applications	execution time and energy
MAUI	2019	[6]	the offloading problem in the edge cloud framework	Single Server	low latency and better energy efficiency	time and better energy
Genetic Algorithm	2020	[7]	task offloading and resource allocation solve the problem of minimizing the overall completion time	Multi-Servers	make the overall completion time shorter	execution time
Heuristic algorithm	2016	[8]	mobile computation offloading problem	Multi-Site	(1) Minimize the energy consumption. (2) Minimize the computation time (3) Minimize the total cost of the computation incurred by cloud computing.	execution time and energy and total cost
DPH and DPR algorithms	2016	[9]	offloading problem	Single and Multi-Servers	completion time and energy consumption	execution time and energy

analyzer retrieves data about program requirements, execution energy costs, and network environment, while the solver determines. The optimization framework dynamically determines whether the method should be offloaded to maximize overall energy savings.

D. Genetic Algorithm

Based Computation Offloading (GACO) [7] uses a Genetic Algorithm (GA) to offload service workflows through Static analysis and online analysis. [7] Randomly defines the initial population as the fitness function that- is applied to selected chromosomes Perform with minimal energy consumption and time. In the crossover step, offspring are created by comparing the two genes individually: Mobility of parental chromosomes and Sensitivity to errors. In the mutation step, the probability that a gene is silenced is related to its error rate and mobile sensitivity. Genetic algorithm (GA) is one of the most common search algorithms that are used to find optimal or near-optimal solutions to difficult and complex problems [10].

E. Heuristic Algorithm for Multi-Site Computation Offloading

Kumar and Lu demonstrated that computation offloading benefits computation-intensive tasks [8]. To improve the performance of mobile computing through the cloud, computing tasks can be outsourced to multiple clouds. A major motivation for multi-cloud is the possibility of offering different prices for different services (e.g., computing time). Additionally, application designers can aim to achieve different performance goals (e.g., throughput, reliability, and cost). This can be achieved by using resources from cloud providers with different performance capacities and charging prices. Multi-cloud resource allocation also benefits from the optimal combination of computing services from multi-cloud providers.

F. DPH and DPR Algorithms

present two dynamic programming algorithms called DPH, Dynamic Programming with Hamming Distance Termination, and DPR, Dynamic Programming with Randomization. According to [9] Dynamic programming is an optimization method that transforms a complex problem into a series of simpler problems that can be solved interactively and iteratively. The proposed DPH and DPR algorithms introduce randomization. In particular, periodically generate random bit strings of 0s and 1s and use their substrings as they improve the solution, a process similar to genetic optimization. The study also populates dynamic programming tables in creative ways to avoid extra calculations on shared substrings. The results show that the algorithm can find good solutions after a reasonable number of iterations.

III. MOBILE CLOUD COMPUTING

Indeed, Mobile Cloud Computing (MCC) in its simplest form, refers to an infrastructure in which both data storage and data processing occur outside of the mobile device. The MCC describes the transfer of computing power and data storage from mobile phones to the cloud as a new paradigm for mobile applications, where data processing and storage are transferred from mobile devices to powerful and centralized computing platforms in the cloud. These centralized applications are then accessed over a wireless connection based on a thin native client or a web browser on a mobile device. Alternatively, MCC can be defined as a combination of mobile web and Cloud Computing (CC), the latter being the most popular tool for mobile users to access applications and services on the web. In short, MCC provides cloud computing and storage services for mobile users [11].

A. Mobile Cloud Architecture

Computation from the concept of MCC, the overall architecture of MCC can be expressed in Fig. 2, mobile devices are connected through functional interfaces between base stations (links) and networks and mobile devices [12]. The mobile user request and information such as ID and location are transmitted to the central processor, which is connected to the server that provides the mobile network service. Mobile network operators can provide services such as authentication, authorization, and billing to mobile subscribers based on the home agent and subscriber data stored in the database. Afterward, the subscriber request is transmitted to the cloud via the Internet. These services are developed using the concept of utility computing, virtualization, and service-oriented architecture e.g., networking, application, and database servers [13]. [14] Focus on the multi-tier CC architecture. This architecture is often used to demonstrate the effectiveness of CC models in meeting user needs.

IV. ADVANTAGES OF MOBILE CLOUD COMPUTING

Regarding the number users of mobile, it has a lot of properties [14], which could be mentioned in the followings:

A. Extend Battery Life

Battery is one of the major concerns of mobile devices. Several solutions have been proposed to increase CPU perfor-

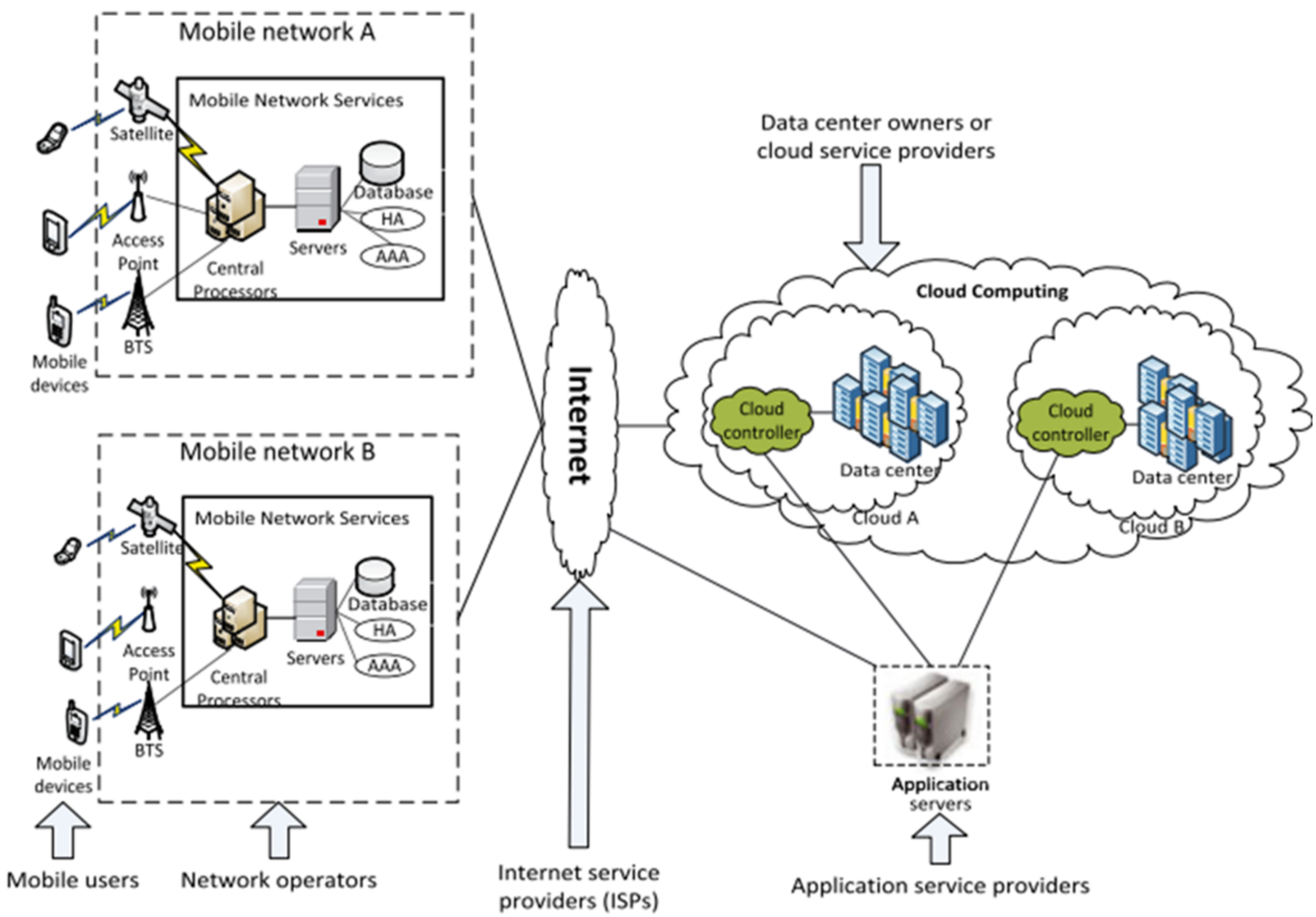

Fig. 2. Mobile Cloud Architecture [14].

mance [13, 14] and manage hard drives and screens to reduce power consumption. However, these solutions require changes to the structure of the mobile device or require new hardware, which leads to increased costs and may not be applicable to all mobile devices.

B. Increase Data Storage

capacity and processing power. Storage capacity is also a limitation of mobile devices. MCC is designed to enable mobile users to store/access large amounts of data in the cloud [15] over wireless networks. The first example is Amazon Simple Storage Service, which supports file storage services. Another example is Image Exchange, which leverages large storage spaces in the cloud for mobile users. This mobile photo-sharing service allows mobile users to upload photos to the cloud immediately after taking them. Users can access all images from any device. Users can save considerable amounts of storage space and energy on their mobile devices.

C. Dynamic Configuration

Dynamic, on-demand provisioning of resources on a self-service, flexible way for service providers and mobile users to run their applications without pre-reserving resources. OPTIMUSCLOUD [16], an online reconfiguration system, can efficiently perform such generic and heterogeneous configurations for dynamic workloads.

D. Scalability

Thanks to flexible resource deployment, mobile application deployments can be accomplished and scaled to meet unpredictable user demands.

Service providers can easily add and scale applications and services with little or no resource usage limitations [17].

E. Multi-tenant

Service providers, such as network operators and data center owners, can share resources and costs to support a wide range of applications and large numbers of users [18].

F. Easy to Integrate

Multiple services from different service providers can be easily integrated through the cloud and network to meet user needs [19].

V. Offloading Objectives

Many studies have investigated how to download data or computationally intensive programs from resource-hungry mobile devices. All surveys target different important factors such as bandwidth, routing, etc. and different objects such as energy consumption, response time, etc. According to their research objectives, these works can be roughly divided into three categories, as in the followings:

A. Saving Energy

Expand Due to the limited battery capacity, battery life is one of the most important design goals for mobile devices. So, in [20] a study for energy-aware high-performance computing has been presented, where they created an energy model to approximate energy consumption during discharge. Studied the feasibility of offloading for mobile computing. It proposed a feasibility assessment method and evaluated the cost of non-cloning and reverse-cloning in terms of bandwidth and power consumption. Created a network coding scheme for mobile cloud computing, which can achieve lower power consumption by reducing the power consumption of CPU and wireless network interface cards. It suggested novel routing methods for efficiently transmitting data in order to optimize the power of nodes. All of the above studies are aimed at maximizing the battery life of mobile devices. There are two models are widely used for the energy consumption one model is based on the Dynamic voltage and frequency scaling (DVFS) [21, 22] that the server energy consumption is linear to the CPU utilization ratio, modeled the energy consumption of the server at the MEC host based on CPU utilization as in (1) [22], assuming a fixed running frequency,

$$E_s = \; a\, E_{max} + (1-a)\, E_{max}\, u \tag{1}$$

Where E_{max} is the energy consumption for a fully utilized server, α is the fraction of the idle energy consumption and u denotes the CPU utilization ratio. Idle power consumption is mainly due to the power consumption in the power delivery and cooling infrastructure.

B. Response Time

Application responsiveness is important, especially for real-time and user-interactive applications. Developed an offloading middleware that provides runtime offloading services to improve the responsiveness of mobile devices. An exhaustive search algorithm was studied to examine all possible application partitions to find the optimal swap partition. All partitioning methods are well suited for small applications. The partitioning problem of multi-user computing is studied, which considers the partitioning of multi-user computing and the scheduling of offloaded computing on cloud resources. The aim is to study transport mechanisms or protocols to solve application transport problems for real-time applications [23].

C. Energy and Time Saving

Energy consumption and response time are two important performance metrics for mobile applications. However, few works address both goals simultaneously. Implemented a framework called ThinkAir [24], that allows developers to easily migrate their application workloads to the cloud. A game-theoretic approach is proposed for efficient computation offloading to optimize execution cost. All these works examine the energy consumption and response time of mobile cloud computing in different ways, providing many important references.

VI. Metrics Used to Make Offloading Decision

The main equation used to find the minimum cost of energy and time used to make a decision of which task process locally or remotely is given by (2) [23] as follows:

$$\min \sum_{i=1}^{s\prime} El_i M_i + EC_i (1 - M_i) \zeta \max_{TL(M_i), TC(1-M_i)} \tag{2}$$

$s.t\ M_i \in \{0,1\}, \forall_i \in \{1,2\ldots,S\}$

When $M = [M_1\ , M_2\ , \ldots, M_s\ ]$ as the vector of binary offloading decisions for set of tasks and EC_i Processing cost, El_i local energy, TL Time to execute task i locally and TC delay for tasks i offloaded to the cloud. The factor ζ is a weighting factor that can be adjusted to change the emphasis on execution delay and energy consumption.

A. Execution Time

Time that offloaded tasks need to execute in Mobile (locally) Tm. Time that offloaded tasks need to execute in cloud server TC in (3) [9].

$$TC\ =\ Tt\ + Tc \tag{3}$$

Where Tt is transmission time and Tc is execution time for specific tasks.

B. Energy

Calculate Energy of cloud server in (4) [9].

$$EC = \ Et\ +\ \beta Cc \tag{4}$$

$$Et\ =\ 0.142\ DI\ +\ 0.142\ DO \tag{5}$$

When β being the relative weight of the cloud energy cost, Cc. Et represents the mobile device transmission energy for both upload and download and it is set to $1.42 \times 10 - 7J/bit$, as in (5). Table II shows list of symbols used in the equations below [9].

TABLE II.
List of Symbols

Symbol	
DI	Input data
DO	Output data
Cc	Cloud Energy
Tt	transmit time
EL /TL	Energy/time to execute tasks locally
EC /TC	Energy/time to execute tasks in cloud
R	transmission rate (R) which is measured in megabits per second
Et	Energy used to transmit data

C. Transmission time

The transmission time, Tt of each task between the mobile and cloud server is equal to the size of data (input/output) for each task divided by the transmission rate (R) which is measured in megabits per second in (6) [9].

$$Tt = \frac{DI}{R} + \frac{DO}{R} \tag{6}$$

VII. Partial and Full Offloading

Most previous research has focused on partial offloading techniques, where resource-critical parts of tasks are offloaded to MCC or MEC servers. These experiments use hybrid offloading techniques and local computation based on dynamic channel conditions. However, a mixed approach of local processing and partial outsourcing does not always work. In particular, the choice between partial and local computation depends on the parameters of the system, e.g., the number of bits to be computed remotely from the computation server [25]. Similar to the static fading condition, full discharge greatly maximizes the computational rate performance of MEC and MCC.

VIII. Mobile Computing (MC)

MC includes mobile communications, mobile hardware, and mobile software. So, the communication topics include adhoc and infrastructure networks, as well as communication properties, protocols, data formats, and specific techniques. Thereafter, the hardware includes mobile devices or device components. The mobile software deals with the characteristics and requirements of mobile applications [26]. Nowadays, mobile devices such as smartphones and tablets are increasingly becoming an integral part of human life, becoming the most efficient and convenient means of communication regardless of time and place. Mobile users get rich experience with various services from mobile device applications such as iPhone apps and Google apps that are running on the device and/or on remote servers over wireless networks. Consequently, the rapid development of MC [27] is becoming a powerful trend in the development of IT technology as well as business and industrial fields.

IX. Mobile Edge Computing (MEC)

Cloud and Mobile Edge Computing (MEC) provides a wide range of computing services for mobile applications [28]. Especially mobile edge computing, where computing and storage infrastructure can be deployed at the edge of the cellular network close to end users. Deploy small cells to build a mobile edge network that can coexist with cloud infrastructure. A large number of businesses and individuals rely on services provided by mobile edge and cloud for their computing and storage needs. Based on user behavior and needs, computing tasks are first offloaded from mobile users to the mobile edge network and then executed at one or more specific base stations in the mobile edge network. MEC architecture is capable of handling a large number of devices, which generate a large amount of traffic. Edge servers are the second layer in be expressed in Fig. 3. Mobile Cloud Architecture [29].

X. Fog Computing System

The ever-increasing computing demands of mobile applications. It can help mobile devices overcome resource constraints by offloading computationally intensive tasks to cloud servers. The cloud challenge is to minimize data transfer and task completion time for users whose location changes due to the mobility and energy consumption of mobile devices. Providing satisfactory computing performance is a particular challenge in fog computing environments. Fog computing is an emerging cloud computing paradigm designed to meet the growing computing demands of mobile applications [30]. It can help mobile devices overcome resource constraints by offloading computationally intensive tasks to remote cloud servers. Fog computing has emerged to provide computation and storage near the data sources [31].

XI. The MCC Applications

Indeed, MCC applications are taking an increasing share in the global mobile market. However, various mobile applications take advantage of MCC. This section describes some types of MCC applications [32]. The end of the 21st century (2020) will be a connected digital world with 8 billion connected devices and 100 billion devices with connections, making possibilities of anything being connected starting from cell phones [33].

1. Mobile commerce (m-commerce) is a business model in which transactions are conducted through mobile devices. Mobile commerce applications typically perform tasks that require mobility (for example, mobile transactions and payments, mobile messaging, and mobile ticketing). Mobile commerce applications can be

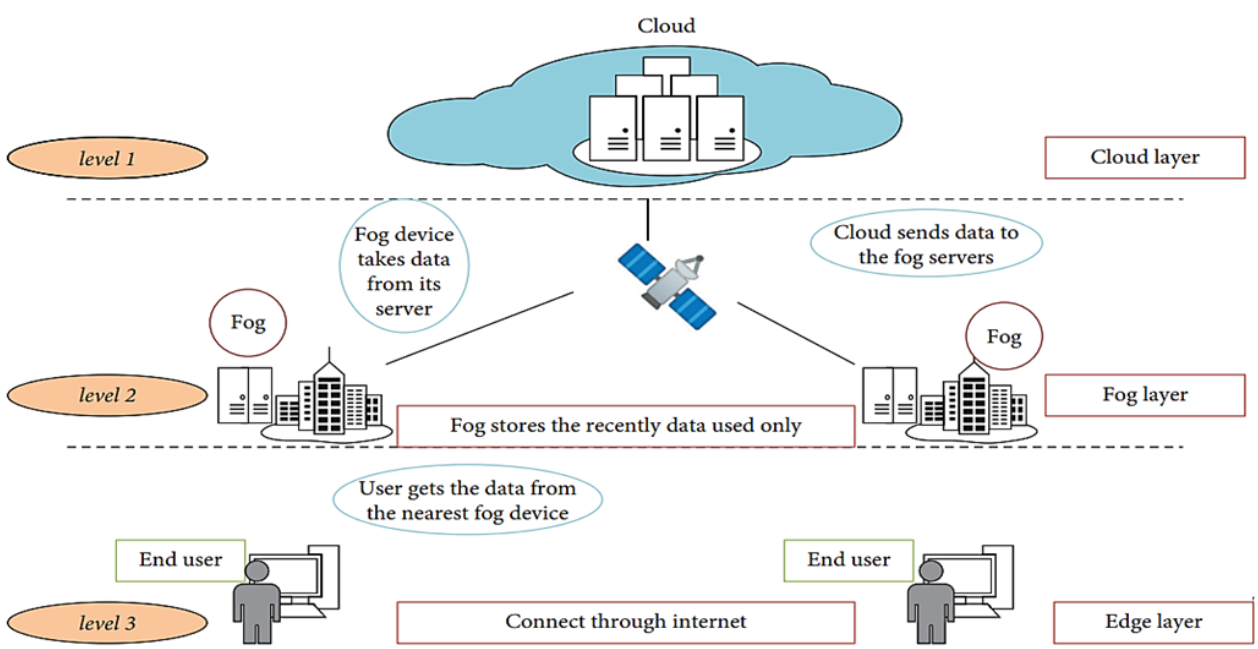

Fig. 3. Layer of cloud [11].

divided into several categories, including financial, advertising, and shopping [34].

2. Mobile games (M-Game) are a potential revenue-generating market for service providers. M-Game can completely offload the game engine requires a lot of computing resources to the cloud server, and players only interact with the screen interface of the device [34].
3. Mobile healthcare (mHealth): The purpose of using MCC in medical applications is to minimize Traditional healthcare, small physical storage, security and privacy, and medical errors. Mobile Healthcare (M-Healthcare) provides mobile users with practical assistance for easy and efficient access to resources such as patient records [35]. Also, mHealth provides hospitals and healthcare organizations with on-demand services in various clouds instead of having standalone applications on local servers. There are a few schemes of MCC applications in healthcare. For example, [36] a mobile application that can assist community pharmacists in the diagnosis and management of minor ailments.
4. Mobile Learning: Mobile learning (m-learning) is designed based on electronic learning (e-learning) and mobility. However, traditional mobile learning applications have limitations such as high equipment and network costs, low network transmission rates, and limited educational resources. Cloud-based mobile learning applications are being introduced to address these limitations [37].

XII. Computing Side Issues

Compute offload. As mentioned in the previous section, offloading is one of the main functions of MCC, which can improve the battery life of mobile devices and improve application performance. However, there are related issues, efficient and dynamic offloading when the environment changes.

A. Offloading in A Static Environment

It turns out that offloading codes are not always effective in saving energy [38]. In code compilation, offloading consumes more power than local processing when the code size is small. A framework was developed by to facilitate task division; one of the programs is divided into server and client tasks. Under our execution model, tasks statically correspond to procedure (or function) call sites. Dynamically, a task is a single call to the appropriate task program. After partitioning, all tasks associated with the same host can share the program state as in the original program. However, consistency issues arise when dealing with data that is shared between two tasks and ends up on different hosts. Data, if the program can only be shared by two hosts Transformation and message delivery, can be guaranteed correct dependency.

B. Offloading in A Dynamic Environment

A dynamic Offload environment can cause these other issues caused by changing connection state and bandwidth. Offload Performance Analysis in Wireless [18] Environments examines three scenarios when running an application, thereby estimating the offload efficiency. These are the cases where the application is running locally (without Uninstall), executed in an ideal unload system (no errors), and executed in the presence of unmount and error recovery. In the last case, the application will be swapped out again when an error occurs. This approach is just re-sourcing what failed Subtasks, which reduces execution time. However, this solution has some limitations. That is, the mobile environment is considered an ad-hoc WLAN (does not support broadband connections). Also, during Offloading, disconnection of the mobile device is considered an error.

XIII. Conclusions and Future Scope

Mobile cloud computing is one of the mobile technology trends, since it compiles the advantages of both mobile computing and cloud computing. Cloud computing has carried many benefits to the computing world. Along with these benefits. The overview has been performed about cloud computing and its applications, as well as mobile cloud computing. It is essential to keep in mind that the designing of future framework solutions should be more cost-effective and should provide security against unauthorized access. Most of the current research exists to make offloading decisions to improve performance and power saving, when offloading decisions depend on metrics, network bandwidths, and amounts of data exchanged between servers and mobile, when the mobile processor speed is almost close to the cloud server processor speed the decision tends to execute the task locally. When the mobile processor speed is slightly lower than the cloud server processor speed and consider that we ignore the data amount and transmission rate the decision tends to execute the task remotely. The critical case is when the cloud server processor speed is better than mobile speed and takes into account data amount and transmission rate in this case the decision is affected a lot by data amount and transmission rate. The growth in task numbers with it is requirements (resources and data) a meta-heuristic algorithm is the solution to offloading decisions for avoiding a limitation in end devices.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

REFERENCES

[1] Q. Xia, W. Liang, Z. Xu, and B. Zhou, "Online algorithms for location-aware task offloading in two-tiered mobile cloud environments," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, (London, UK), pp. 109–116, IEEE, 2014.

[2] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *Journal of Network and Computer Applications*, vol. 169, p. 102781, 2020.

[3] M. Maray, J. Shuja, *et al.*, "Computation offloading in mobile cloud computing and mobile edge computing: survey, taxonomy, and open issues," *Mobile Information Systems*, vol. 2022, 2022.

[4] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *2012 Proceedings IEEE Infocom*, (Orlando, FL), pp. 945–953, IEEE, 2012.

[5] J. L. D. Neto, S.-Y. Yu, D. F. Macedo, J. M. S. Nogueira, R. Langar, and S. Secci, "Uloof: A user level online offloading framework for mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2660–2674, 2018.

[6] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: Issues, methods, and perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–23, 2019.

[7] Z. Li and Q. Zhu, "Genetic algorithm-based optimization of offloading and resource allocation in mobile-edge computing," *Information*, vol. 11, no. 2, p. 83, 2020.

[8] N. I. M. Enzai and M. Tang, "A heuristic algorithm for multi-site computation offloading in mobile cloud computing," *Procedia Computer Science*, vol. 80, pp. 1232–1241, 2016.

[9] H. Shahzad and T. H. Szymanski, "A dynamic programming offloading algorithm for mobile cloud computing," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, (Vancouver, BC, Canada), pp. 1–5, IEEE, 2016.

[10] R. J. Essa, N. A. Abdulah, and R. D. Al-Dabbagh, "Steganography technique using genetic algorithm," *Iraqi Journal of Science*, vol. 59, no. 3A, pp. 1312–1325, 2018.

[11] M. G. Jaatun, G. Zhao, and C. Rong, *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009, Proceedings*, vol. 5931. Berlin: Springer Science & Business Media, 2009.

[12] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[13] H. R. Nikkhah, R. Sabherwal, and J. Sarabadani, "Mobile cloud computing apps and information disclosure: the moderating roles of dispositional and behaviour-based traits," *Behaviour & Information Technology*, vol. 41, no. 13, pp. 2745–2761, 2022.

[14] S. S. Woo and J. Mirkovic, "Optimal application allocation on multiple public clouds," *Computer Networks*, vol. 68, pp. 138–148, 2014.

[15] Amazon, "Amazon s3: Object storage built to retrieve any amount of data from anywhere," 2024. https://aws.amazon.com/s3/.

[16] A. Mahgoub, A. M. Medoff, R. Kumar, S. Mitra, A. Klimovic, S. Chaterji, and S. Bagchi, "{OPTIMUSCLOUD}: Heterogeneous configuration optimization for distributed databases in the cloud," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pp. 189–203, 2020.

[17] D. K. Viswanath, S. Kusuma, S. K. Gupta, *et al.*, "Cloud computing issues and benefits modern education," *Global Journal of Computer Science and Technology*, vol. 12, no. 10, pp. 15–19, 2012.

[18] V. Narasayya and S. Chaudhuri, "Multi-tenant cloud data services: state-of-the-art, challenges and opportunities," in *Proceedings of the 2022 International Conference on Management of Data*, (Philadelphia, USA), pp. 2465–2473, 2022.

[19] X. Yang, T. Pan, and J. Shen, "On 3g mobile e-commerce platform based on cloud computing," in *2010 3rd IEEE International Conference on Ubi-Media Computing*, (Jinhua, China), pp. 198–201, IEEE, 2010.

[20] M. Wolf, *The physics of computing*. Cambridge, US: Elsevier, 2016.

[21] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-efficient virtual machine provision algorithms for cloud systems," in *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, (Melbourne, VIC, Australia), pp. 81–88, IEEE, 2011.

[22] S. Thananjeyan, C. A. Chan, E. Wong, and A. Nirmalathas, "Mobility-aware energy optimization in hosts selection for computation offloading in multi-access edge computing," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1056–1065, 2020.

[23] J. Wang, J. Peng, Y. Wei, D. Liu, and J. Fu, "Adaptive application offloading decision and transmission scheduling for mobile cloud computing," *China Communications*, vol. 14, no. 3, pp. 169–181, 2017.

[24] S. Jeyalaksshmi, J. A. Smiles, D. Akila, D. Mukherjee, and A. J. Obaid, "Energy-efficient load balancing technique to optimize average response time and data center processing time in cloud computing environment," *Journal of Physics: Conference Series*, vol. 1963, no. 1, p. 012145, 2021.

[25] A. Andrawes, R. Nordin, Z. Albataineh, and M. H. Alsharif, "Sustainable delay minimization strategy for mobile edge computing offloading under different network scenarios," *Sustainability*, vol. 13, no. 21, p. 12112, 2021.

[26] I. M. Ali and M. I. Salman, "Sdn-assisted service placement for the iot-based systems in multiple edge servers environment," *Iraqi Journal of Science*, pp. 1525–1540, 2020.

[27] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[28] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. Fitzek, "Device-enhanced mec: Multi-access edge computing (mec) aided by end device computation and caching: A survey," *IEEE Access*, vol. 7, pp. 166079–166108, 2019.

[29] M. N. Abdulredha, A. A. Bara'a, and A. J. Jabir, "An evolutionary algorithm for task scheduling problem in the cloud-fog environment," *Journal of Physics: Conference Series*, vol. 1963, no. 1, p. 012044, 2021.

[30] J. Moura and D. Hutchison, "Fog computing systems: State of the art, research issues and future trends, with a focus on resilience," *Journal of Network and Computer Applications*, vol. 169, p. 102784, 2020.

[31] S. K. Abdullah and A. J. Jabir, "A multi-objective task offloading optimization for vehicular fog computing," *Iraqi Journal of Science*, vol. 63, no. 2, pp. 785–800, 2022.

[32] N. S. Desai, "Mobile cloud computing in business," *International Journal of Information*, vol. 6, no. 1/2, pp. 197–202, 2016.

[33] S. Shamala and A. J. Jabir, "Mobility management for internet of things," *World Scientific News*, vol. 41, p. 313, 2016.

[34] A. Alzahrani, N. Alalwan, and M. Sarrab, "Mobile cloud computing: advantage, disadvantage and open challenge," in *Proceedings of the 7th Euro American Conference on Telematics and Information Systems*, pp. 1–4, 2014.

[35] P. K. D. Pramanik, B. K. Upadhyaya, S. Pal, and T. Pal, "Internet of things, smart sensors, and pervasive systems: Enabling connected and pervasive healthcare," in *Healthcare data analytics and management*, pp. 1–58, Amsterdam, Netherlands: Elsevier, 2019.

[36] E. M. Mikhael, F. Y. Al-Hamadani, and A. M. Hadi, "Design and evaluation of a new mobile application to improve the management of minor ailments: a pilot study," *BMC Health Services Research*, vol. 22, no. 1, p. 920, 2022.

[37] I. M. Al-Joboury and E. H. Hemiary, "Internet of things architecture based cloud for healthcare," *Iraqi Journal of Information and Communication Technology*, vol. 1, no. 1, pp. 18–26, 2018.

[38] P. K. D. Pramanik, N. Sinhababu, B. Mukherjee, S. Padmanaban, A. Maity, B. K. Upadhyaya, J. B. Holm-Nielsen, and P. Choudhury, "Power consumption analysis, measurement, management, and issues: A state-of-the-art review of smartphone battery and energy usage," *IEEE Access*, vol. 7, pp. 182113–182172, 2019.