STM JOURNALS

International Journal of
Mobile Computing Technology

ISSN: 3049-0871
Volume 3, Issue 2, 2025
July—December
DOI (Journal): 10.37591/IJMCT

https://journals.stmjournals.com/ijmct

Review IJMCT

Offloading Computation to the Cloud

V. Basil Hans*

Abstract

Mobile devices are increasingly relied upon for complex and resource-intensive applications such as real-time video processing, augmented reality, and machine learning. However, their limited computational power, storage capacity, and battery life pose significant challenges. Computation offloading to the cloud has emerged as a promising solution to overcome these limitations by transferring demanding tasks from mobile devices to remote cloud servers. This approach enables improved performance, reduced energy consumption, and enhanced user experience. The study discusses various computation offloading models, including full, partial, and dynamic offloading, and explores the role of edge and fog computing in minimizing latency and improving real-time responsiveness. It also talks about important problems like security, network dependability, cost-effectiveness, and decision-making algorithms for the best offloading. The study concludes that integrating intelligent offloading strategies with 5G and edge computing technologies can pave the way for a new generation of efficient and adaptive mobile applications.

Keywords: Mobile cloud computing, computation offloading, edge computing, energy efficiency, latency reduction, resource management, 5G networks

INTRODUCTION

Cloud offloading refers to the transfer of computation and/or storage tasks from user devices to cloud-based infrastructure. This approach employs high-performance computing facilities in the cloud to complement resource-constrained devices, enabling users to take full advantage of cloud capabilities. The core of cloud offloading involves the cooperative sharing of tasks between devices and the cloud. Users decide whether to perform operations on the local device or offload them to a cloud server, based on the characteristics of the tasks, to maximize the performance of their applications, while still ensuring Quality of Service (QoS).

Cloud offloading is usually a response to the increasing complexity of applications that cannot meet required performance thresholds under local execution. The evolution of applications also drives the need for cloud offloading. For instance, the rapidly growing popularity of deep learning-based image processing applications for augmented reality, such as Prisma and Snapseed, leads to ever-growing burdens on mobile devices. Therefore, the requirement for cloud offloading is becoming more and more crucial.

***Author for Correspondence**
V. Basil Hans
E-mail: vhans2011@gmail.com

Research Professor, Department of Management & Commerce, Srinivas University, Mangalore, Karnataka, India

Received Date: October 26, 2025
Accepted Date: October 27, 2025
Published Date: November 01, 2025

Citation: V. Basil Hans. Offloading Computation to the Cloud. International Journal of Mobile Computing Technology. 2025; 3(2): 27-37p.

The first challenge of cloud offloading is to decide whether to offload a composite task to the cloud. The second challenge is to identify the offloading strategy for the task if it is determined that partial or entire offloading is necessary. Complex applications, such as video processing and image recognition, often comprise several subtasks that can be executed in sequence or parallel. It is not easy to decide how many subtasks to offload, how to do the offloading (for example, first stage local, second stage cloud; first stage cloud, second stage

local), and what data to send through bandwidth-limited channels between the mobile device and the cloud while the application is running [1–3].

MOTIVATION FOR CLOUD OFFLOADING

Offloading computation to the cloud (Offload) empowers devices to distribute computational tasks requiring large resources outside their own local environments onto nearby, powerful, and resource-provisioned Cloud Computing Services (CCS). Offloading brings significant advantages in terms of performance, scalability, and energy favorability over executing the computation entirely on edge devices. Due to the advanced performance, energy, and time efficiency of CCS and progressing infrastructure, offloading techniques now not only apply to large-scale workloads such as PDA tasks but also data-associated tasks like extraction of features from captured photographs [1]. Hence, an outline of typical workloads that can benefit from offloading is a necessity.

Consider the performance gains brought by Offloading. Edge devices are often restricted by processing capability, battery of operation, memory storage, and the time to reach a decision. Cloud Systems (CS) in the vicinity increase the performance capability and enhance the overall system execution. Offloading from portable devices to nearby clouds potentially reduces the computation time tremendously while the portable device stays the same. Scale is another factor; large-scale ultra-hyper-parameter workloads cannot be well accommodated by portable devices, offloading for a CS that contains 100s of CPU-GPU-TPU accelerators becomes essential. Offloading also grants an energy reservation effect. Offloading and computing on devices carrying large-parameter solutions consume the battery quickly. Offloading to processing hubs allows saving energy and lengthening the device time on the work with milliwatt-level consumption. The cloud connections affect the cost point. For some companies that want to plan the Fulfilment of Operation (FOOs), offloading to CCS is cheap and helpful. However, when employees work from home and only use it 10–20% of the time, the monthly operation expenditure (OE) will be 50000–60000×10^(-4) dollars. The similar scoped but never-off-loaded workloads will cost more.

ARCHITECTURAL CONSIDERATIONS

Cloud offloading offloads computations from a client to cloud-based resources to improve end-user experience, energy efficiency, and resource management. Although cloud computing started with stationary machines, the widespread adoption of mobile, wearable, and IoT devices has motivated a cloud computing paradigm where computations from such devices can also be offloaded to remote cloud resources, thereby providing several benefits [3]. In a mobile environment, wireless channels connecting the client and the cloud are subject to fading, interference, and handover, which gives rise to various degrees of disconnection and data loss [4]. Thus, a cloud-centric computing paradigm where computations from mobile, wearable, or IoT devices are offloaded to any distant cloud decreases traffic and enhances user experience, device lifetime, and battery saving [5].

Mobile edge computing refers to a distributed computing paradigm in which computation, storage, and networking resources are moved toward the edge of the network to cover cloud service resources as close as possible to end users. Originally, mobile edge computing was created to get video files with good quality over limited bandwidth. It can now also include mobile cloud computing platforms, where computations from mobile devices are done in the cloud to cut down on the amount of data sent over wireless channels that do not have a lot of bandwidth. Although mobile computation offloading shows significant improvement in battery saving, only part of the computations from mobile terminals can be offloaded because some computations require local data, access control, and/or real-time response.

Client-Cloud Interaction

Recent developments in cloud computing technologies have opened the door to a new generation of computing paradigms in which processing is performed remotely on powerful servers. This and other models fall under the umbrella term cloud offloading, defined as the process of transferring application

operations and data loading from a local mobile device to a distant cloud server [6]. An offloaded application thus leverages cloud resources to reduce local energy consumption, access large data sets, and perform heavy computations without the need for a native and local implementation. Heavy workloads in on-device programming incur substantial energy expenditure and runtime delay, leading to the gradual consideration of cloud resources for computation [3]. The candidate programs that are viable for cloud offloading are those that run for prolonged periods, consume large amounts of energy, or make extensive use of computations and data accesses during their course of execution. Cloud offloading reduces client device energy consumption by transferring both computation and data to the cloud for execution. Applications that require massive data processing can benefit from cloud resources-enabled data offloading.

Data Management and Security

Offloading computation to the cloud remains unpopular in mobile devices, although the range of computationally demanding applications is increasing in sectors such as mobile gaming and augmented. Among several approaches for offloading, transferring data to a cloud server to perform elaborate computations and return results is widely adopted. Offloading data brings undeniable benefits in terms of computation speed, battery efficiency, cost, and energy consumption. Many cloud service providers, such as Amazon, offer free-tier services with limited resources, enabling cost-free explorations for mobile offloading development. Cloud computing enforces careful evaluation of resource availability and the associated fees. The elasticity of cloud resources leads to challenges in estimating high watermark consumption, resulting in unpredictable bills. The following section shows a wide range of applications, workloads, and tasks that manifest in mobile systems.

Offloading data to the cloud involves specific cloud acquisition and management processes. Different applications vary in sensitivity along a wide spectrum, encompassing issues such as data privacy, user profiles, and domain knowledge. The variety of data nature requires careful attention to categories such as textual, visual, speech, acoustic, and programmatic. The ownership of datasets collected may differ for academia, industry, and personal conduct, forming another dimension demanding specialized measures. Systematic categorizations enable practical key observations around adaptable configurations or guidelines for different scenarios, forming the basis of stringent regulations imposed upon function provided by the target cloud.

Cloud storage has become a widely adopted service aimed at fulfilling the necessity of data availability. Users often prefer configuring cloud storage by synchronizing content of target devices with the cloud when data growth turns excessive on mobile devices within limited storage resource. Offloading selective datasets keeps scholars abreast of emerging changes without fully grasping the entire dataset. Different users in the cloud possess distinct privacy requirements, yet measures for safeguarding data remain standard rather than target-dependent [7].

Latency and Bandwidth Implications

Applications that require real-time interaction need to bound the end-to-end delay that users observe. For offloaded invocations, there is an additional delay called network latency. The combination of local and network delays helps define an overall cloud offloading framework through Quality of Service (QoS) metrics. Offloading requires that at least one network connection is available to the mobile device and that offloading for any task is triggered based on available network bandwidth and delay. In mobile-to-cloud environments, both the base stations and mobile devices are capable of fast handover and have low latency connections. State-of-the-art infrastructure supporting transmission on the order of milliseconds is thus prevalent.

WORKLOAD PARTITIONING AND OFFLOAD STRATEGIES

Partitioning is the process of selecting specific computation nodes in executing various applications, from cloud-based distributed supercomputers to handheld devices. In devising static workload

partitioning and offloading strategies, developers need to determine the boundary of data exchange points between the client and cloud to perform desired computations at both sites. The selection of fixed partition points depends on application design parameters, communication cost for transfers, and cloud server availability [6]. Offloading decisions must be correctly examined and verified during the analysis since erroneous criteria may incur high overhead and waste resources. The completion time must be known for both client and cloud computations. This allows the cloud candidate selection to take place and identifies the offloading and workload-sharing scenarios, alongside ways of assembling outputs after execution.

The goal of adaptive offloading is to reduce latency while minimizing resource consumption for the mobile device. Existing solutions for policy-based workload partitioning cannot satisfy requirements under such circumstances due to variations in offloading prerequisites during runtime, meaning they lack the support for mobile applications. During the execution of a particular application, the device records multiple parameters which can be used to specify the execution constraints associated with previously completed offloading and to formulate execution time estimation models, assisting re-partitioning decisions, for the on-going application. Load balancing among the cloud nodes is also taken into consideration to avoid congestion.

Static Partitioning

Cloud-based applications must execute assignment, data, and control precedence throughout program execution to improve application performance, conserve energy, maintain connectivity, and minimize operating cost on mobile nodes. Careful selection of offloading tasks enhances application execution by ensuring work is assigned to the appropriate device. Static partitioned applications and cloud-based services frequently deploy systems without explicit or implicit data dependencies, ensuring data can be preloaded to mobile nodes and information propagates quickly and predictably from mobile devices to location-based services [8]. Static partitioned applications use a single offloading scheme and local clients execute tasks independently on local data without reliance upon the cloud and required application service.

Dynamic and Adaptive Offloading

Offloading computation-intense tasks to the cloud allows energy-efficient and ultra-high-performance mobile applications. A dynamic and adaptive offloading engine intelligently selects between three execution modes: cloud, on-device, or a combination thereof, based on context. A runtime-monitor system systematically collects context data and uses it to determine the offload mode [1]. Furthermore, the system comes with a sophisticated feedback mechanism, which re-evaluates offloading decisions at run-time and supports both global and local re-partitioning strategies depending on context variations.

Granularity of Offload

Granularity, a key aspect defining cloud workload offloading, typically refers to the size of data exchanged and the number of associated operations performed [2]. Offload decisions may be made at coarse or fine granularity. A coarse-grained approach requires the transfer of entire data objects such as files, whereas a fine-grained one transfers only small data portions such as records or tuples. Coarse granularity can lead to significant data movements subject to large serialization overheads. Conditional on input data size, offloading functions may exhibit either a coarse- or the fine-grained pattern when data locality, serialization overhead, and cache effects are also taken into consideration [3]. Relevant drivers encompass variable input sizes, data locality, serialization overheads, and caching effects.

COST AND RESOURCE MANAGEMENT

When cloud resources incur charges based on actual usage, an explicit distinction corresponds between a rent-as-you-go model [9] and contract-based reservation. The first case exhibits cash-flow varying jointly with activity levels, influencing time-multiplexing of routed loads on the cloud. The

latter model maintains a constant subsystem cost, where forecasts of minimum engaged services allow deriving cash flow distribution patterns. As push-and-pull cloud pricing approaches evolved, various anticipated-utility curves plotting performance quotients and task execution outlay surfaced across such communities, including peer-to-peer networks, grid computations, and file sharing applications.

Such guidelines establish a mapping between the eventual offloading-delineation scheme and the expected finish-event duration and shape an endeavor on commodity and resource-equipped remote machines that remains triggered by price-active estimations.

When projected workloads and corresponding remunerations remain inside pre-assigned asset-limits, a cloud counterpart can own a rise in quantity. In conjunction, the common executing length of the preserved program resides in anticipation of appropriate operation of access and pursued specifications, yet virtually every passing provision must discuss security, availability, and coverage-support sectors among apparatuses [10].

Pay-as-you-go versus Reserved Models

The pay-as-you-go pricing model is the foundation of cloud computing. A user only needs to pay when a resource is actually used instead of committing funds upfront, as in the case of "reserved" resources. Such subscriptions offer a discount compared to pay-as-you-go prices but require an often-intuitive understanding of future resource demand and computing workload schedule to minimize overall costs. Whether purchasing reserved resources is more efficient than leasing on-demand resources depends on the prediction accuracy of the resource demand curve [11]. Generally, the penetration rate of paid resources (such as social media and video weather data collection) obtained from numerous companies and/or government databases can be directly tracked from the web, while energy-consumption to resource-utilization models are rarely available. However, using monthly and hourly historical paid-resource information with equal time-period ranges does provide some assistance regarding how to calculate expected-resource demand in cloud computing [12].

Performance versus Cost Trade-offs

Most cloud services charge on a pay-as-you-go basis for service usage per time duration or number of requests [4]. Benefits include no need for upfront investment, immediate scaling without exposure to long-term risk, and control of cash outflow. In contrast, reserved payment provides lower unit costs in exchange for long-term financial commitment [13]. Offloading decisions are thus complicated by the need to manage cost alongside performance. Time consumed by the application generally translates directly to cost, but a more complex relationship exists because base-state time governs subsequent input processing. If performance improves during execution, validation times can also decrease with an imperfect model. If cloud service engages in parallelism or utilizes hardware acceleration also present at the client, the client transformation stage can become a distinct bottleneck.

SECURITY, PRIVACY, AND COMPLIANCE

Cloud environments depend on accessibility to high-volume and high-velocity data. There are diverse general-purpose management tools to create, process, manipulate, archive, destroy, and retrieve data in the cloud. Yet users may still feel apprehensive about transferring data to a third-party provider. Robust data protection is therefore essential to compliance with various regulations, especially in terms of location, movement, access, and control. Some regulations govern personal data storage and use in cloud computing. Such data may be sensitive or confidential; safeguards can establish a higher level of trust and assurance through risk control, theft prevention, and malware protection; and creditors, insurers, and associates may require advanced measures to satisfy regulations or corporate policies before accepting a revenue or revenue-generating partner [14]. Administration of models, images, and clusters deployed on external servers agrees with relevant specifications [15], and specific guidelines pertain to the protection of sensitive, personal, and BRITE-target data [16].

Encryption and Key Management

A cloud service typically transmits the contents of files between a storage location and a client. Thus, files must be encrypted while they are stored and when they are transmitted to a cloud service. Hence, two essential encryption practices help maintain data confidentiality: encrypting files before transferring them to cloud storage, commonly known as "encryption at rest", and encrypting files before transmission begins, called "encryption in transit" [17]. Such encryption practices help provide security against potential breaches, as another restriction would be put on data users.

Following the completion of file encryption during the upload process, the appropriate solution to files to be stored in cloud services requires examination. A persistent key allowing encrypted files to be uploaded to a storage location would still require file decryption in situations where a client needs to access these files. These restrictions produce the fundamental key cycle as follows: file encryption, storage in cloud services, and file decryption. Following this cycle, the key generation and maintenance model becomes apparent: a key needs to be generated and managed following each storage, thereby forming the key lifecycle [18].

Access Controls and Auditing

The cloud paradigm allows application developers to offload computation to non-local resources. Precious on-device resources, such as battery energy and CPU cycles, may thus be gained. Offloading is possible owing to the highly interconnected society in which humans now live; devices predominantly remain connected to the Internet via wireless access points, be they cellular or Wi-Fi, at nearly all times.

The major driving forces behind cloud offloading include performance gain, enhanced scalability, increased energy efficiency, lower cost, and the presence of well-defined criteria that determine the offloading and on-device execution requirements. Regarding performance, many workloads have certain computational and transmission characteristics that can exploit cloud resources via cloud offloading. Such workloads can thus complete faster via cloud execution rather than on-device execution. Offloading reduces energy consumption and active time at the expense of cloud resource and uploading energy consumption. During cloud execution, the device primarily consumes transmission energy to upload input data and download output results [19]. Cloud frameworks follow diverse charging schemas, and an offloading decision simplifies the substantive cash-flow forecasting and utilization metric by determining how much computation and networking to undertake. The cloud can support workloads requiring bursts of large amounts of processing, large and unpredictable transaction sizes, and large portions of stateful interactions and diverse data formats.

Compliance Considerations

Cloud computing is regulated in several areas and among those areas cloud service compliance is gaining importance. When organizations subscribe to cloud services one of the technical aspects of compliance that must be considered is the ability for the organization to maintain Governance and Control. Cloud services if not properly implemented, can result in loss of information visibility, which is one of the main requirements for PCI DSS. With cloud computing, data is kept on a remote server instead of directly on the computer you have access to at all times. Information regarding data location (cloud services use multi-tenancy and security separation, and information visibility aspects) must be known by the organization.

Organizations need to first know what regulations or standards apply to them before considering cloud services. The National Institute of Standards and Technology (NIST) has published several documents regarding cloud computing, including security standards, guidance, and compliance. With regards to compliance guidelines, the service model the organization is considering must be known. Therefore, differences between IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) must be taken into account [4].

RELIABILITY, FAULT TOLERANCE, AND CONSISTENCY

Cloud solutions provide robustness, high availability, and fault tolerance. Businesses often adopt cloud computing to avoid downtime, repair and maintenance costs, and disgruntled users who suffer interruptions during peak working hours [20]. The frequency of server failures, component failures, and storage failures is usually lower in the cloud than on local hardware. Most cloud systems offer failover mechanisms that allow seamless recovery. When a disaster occurs, the backup configuration file can be used to restore the environment or process.

The cloud architecture includes standby infrastructure in standby mode that does not attract customer requests until a primary failure occurs. Upon customer request, the attached components replicate data when deployment begins for a quick restore of critical information. Every backup infrastructure engages in remote copy activity on a specific partition. Through global checkpointing, multiple client sites can seamlessly recover from software failures.

Failover Mechanisms

Cloud offloading systems must maintain availability and reliability to ensure continuous service. Availability refers to the probability that a service is operational at any given time, while reliability is the probability that a service remains operational without interruption for a specified period. Failures can occur in different domains, so systems must be designed to respond well within those domains. However, it does not guarantee 100% availability. Hence, it is necessary to adopt appropriate operational measures to improve availability [20].

One widely used operational measure to achieve higher availability is a failover mechanism. A failover procedure is a sequence of actions taken in response to a failure during which the system either restores or continues to provide the required service. Failover requires the provisioning of resources in standby mode, allowing the system to continue providing services. Two types of failover are commonly adopted in cloud systems: resource-based failover (standby resources) and process-based failover (restart). The investment of standby resources increases the total cost of the system; thus, more information is needed to adopt an appropriate failover mechanism.

Data Synchronization

Data offloading involves both transmitting data to the cloud for computation and sending back results, with the amount of data sent to the cloud usually exceeding the size of the result returned. Large-scale data-driven mobile applications could improve responsiveness and battery life through cloud data offloading to handle CPU-intensive tasks remotely. Implementing a distributed shared memory system between mobile devices and cloud servers can facilitate scalable and fine-grained offloading by continuously replicating device memory and enable low-latency workload distribution. Offloading decisions depend on performance characteristics (such as device specifications, network bandwidth, and task execution time), application requirements (e.g., either data or computation offloading), execution context, and user preferences.

With increasing reliance on cloud computing, massive amounts of data are stored in various forms across data centers, often stored independently in isolated silos and poorly integrated for cross-referencing. Eventually, an input image may reside in Camera Blob, process scripts on User Script, and related edited images in Image Vertex. Cloud Offloading needs to support state synchronization between these silos and widely distribute synchronization operations among independent services. Data synchronization plays a pivotal role and typically involves three interdependent phenomena: (1) replication, often according to various topologies within heterogeneous distributed systems, (2) conflict resolution, determining a target state upon concurrent update operations across different replicas, and (3) various guarantees on consistency, ensuring state correctness and legal flow of updates across system components.

Consistency Models

A cloud-offloading environment consists of mobile devices, edge servers, and cloud resources. Because mobile devices have significant limitations on computation, energy, storage, and bandwidth, they utilize services from both edge and cloud resources. These services are in a specific state, so the cloud-offloading system must maintain consistency. Data consistency during offloading has been studied extensively, especially on cloud-centric and mobile-cloud computing. To accommodate various applications and technologies, some studies have defined consistency models for data storage within cloud services. The different consistency models for data can be categorized into strong, eventual, causal, and hybrid. Strong consistency guarantees that the result of all read operations reflects the result of some preceding write operation in the system and is the most intuitive for programmers. Public-cloud databases provide strong consistency, but performance degrades linearly with the number of regions. Eventual consistency follows the CAP theorem and is widely adopted. It eventually ensures that all replicas converge to the same state without the need for synchronization, but it may not be suitable for mobile-cloud environments. Hybrid consistency is a mixture of both strong and eventual consistency. Cloud services such as Firebase support hybrid models [21].

EVALUATION AND BENCHMARKS

Outstanding performance and cost-effectiveness remains a driving force for offloading computation to the cloud. Equally important is the large-scale adoption of smartphones, mobile devices embedded with myriad sensors, and the proliferation of Internet of Things (IoT) devices. Opportunities to leverage cloud infrastructure, regardless of the scale of the mobile system, exist in resource-constrained settings, thereby motivating the exploration of a cloud-centric offloading paradigm.

Offloading decisions, supported by operational metrics, depend heavily on workload execution status, device and offload target properties, and user preferences. Partitioning and offload strategies exhibit great variability based on the underlying architecture and available resource configurations. Accordingly, cloud offloading can benefit from recognizing workloads already optimized for the cloud or embodying a bifurcation point between designated on-device and cloud-executable tasks. Measurements wrap the use of scheduled jobs, jobs with waiting time, and predefined time intervals with consideration for alternative resource metrics (e.g., price, quality) and predetermined task sub-portions.

Computation offloading enables devices with limited processing capability to offload computation-intensive tasks to a nearby cloud. A framework that provides a set of performance metrics for monitoring mobile cloud computing has consequently been developed [22]. In addition to offloading, it monitors other Quality-of-Service (QoS) parameters, such as energy and cost [22]. Benchmarks characterize the performance and price of a diverse set of Google applications hosted on various cloud service providers and geographical locations. They further highlight that neither platform-independent nor platform-agnostic benchmark systems exist for mobile cloud services. Benchmarks are vital for comparator systems in networked facilities, and various studies of cloud computing performance actively consider benchmark values.

As devices progress toward the all-seeing and hearing eye, securing their content becomes critical. The challenge of securing the device's content, which transmits via Wi-Fi low-frequency packets, remains unsettled [23]. Cloud benchmarks comprise independent workloads, with no prior knowledge about the respective workload distribution required.

Workload performance in a variety of execution environments (e.g., Local, Cloud, IaaS, PaaS, SaaS, and public versus private clouds) defines an environment-independent benchmark for comparative analysis. While existing benchmarks inform workload selection, system-level knowledge of cloud configuration remains limited at both the applications and network level. Defining performance and cost as multiple objectives provides insights into the characteristics of candidate workloads supporting cloud execution and helps further refine application offloading.

PRACTICAL GUIDELINES AND PATTERNS

Cloud offloading offers potential solutions for computation-hungry applications during periods of high demand when on-device resources cannot keep up. This section summarizes architectural considerations, workload partitioning strategies, cost and security issues, reliability and consistency concerns, and evaluation frameworks in the hope of stimulating wider interest in cloud/offload systems.

Many decision makers need not engage in exhaustive analyses to identify feasible approaches. Several straightforward guidelines capture the most common architectural designs. Typical patterns define the overall system structure, including common partitioning mechanisms, client/device configurations, and relevant trade-offs.

FUTURE TRENDS

Cloud offloading has reached a maturity level, leading to widespread adoption in numerous applications across diverse domains. Nevertheless, important research challenges remain to be addressed and the corresponding survey on the subject helps to highlight these challenges, framing them as ongoing research opportunities. A large number of machine learning services are pushing the machine learning models to be stored and executed in the cloud. As a consequence, one of the new research opportunities will target on-device machine learning services and on-device model learning, as different domains should have such services to protect the domain knowledge and personal privacy. The optimization of multi-device cloud offloading also continues to be an important research direction, particularly for time-sensitive applications that include augmented reality (AR). With the emergences of multi-access edge computing, multimedia cloud gaming and video communication will rise in popularity. Existing techniques related to offloading in and among computing nodes need advancements. Inputs of these applications are videos; and processing latency is critical for video communication. It remains a challenge to jointly optimize video encoding, caching and compute offloading along with video buffering for multiple video streams originated from different devices. Techniques for communication-efficient distributed training will also be highly desirable given the increase in video encoding complexity when dealing with higher resolution and frame-rate [2]. The new trend will be more flexible and capable of dynamically adapting to the different requirements of IoT, real-time and emerging applications [24].

CONCLUSION

Cloud offloading consists of migrating computational tasks from a resource-constrained device to the cloud, thereby alleviating the device's burden [6]. This work presents an overview of cloud offloading solutions for mobile devices, particularly smartphones and IoT equipment. Offloading has received much attention, owing to the restricted processing capacity of mobile devices. Nevertheless, only a limited number of papers solely dedicated to cloud offloading have appeared, demonstrating the scientific relevance of the topic and a significant opportunity to conduct research in this direction. The demand currently exists for a framework that clearly delineates the acknowledged concepts related to offloading applications from mobile devices to the cloud.

Mobile devices have become ingrained in everyday life, driven by the prevalence of mobile applications promoting productivity and entertainment. Smartphones now boast computation power on par with laptops, supporting demanding applications such as gaming, video processing, augmented reality, and complex image processing. The Internet of Things (IoT) has intensified the utility of cloud computing by extending its influence to devices such as wearable systems, cameras, travel trackers, refrigerators, and smoke detectors. When unbounded by processing or battery constraints, the installation period of applications for mobile devices reflects a growing tendency toward further complex software at the expense of local resources. Execution relies increasingly on cloud servers. Since cloud services rely on server accessibility, the high-cost setup of mobile devices that demand computational assistance often leads to potential forfeiture of the initial investment; consequently, accurate offloading decisions based on usage patterns can enhance the overall computational efficiency and assist in weathering the transitional phase between sample and enduring services.

REFERENCES

1. Chen X. Decentralized computation offloading game for mobile cloud computing. IEEE Trans Parallel Distrib Syst. 2014 Apr 11; 26(4): 974–83.
2. Wang J, Pan J, Esposito F, Calyam P, Yang Z, Mohapatra P. Edge cloud offloading algorithms: Issues, methods, and perspectives. ACM Comput Surv. 2019 Feb 13; 52(1): 1–23.
3. Khalili S, Simeone O. Inter-layer per-mobile optimization of cloud mobile computing: a message-passing approach. Trans Emerg Telecommun Technol. 2016 Jun; 27(6): 814–27.
4. Luzuriaga J, Cano JC, Calafate C, Manzoni P. Evaluating computation offloading trade-offs in mobile cloud computing: A sample application. In Proc 4th Int Conf Cloud Comput, GRIDs, Virtualization. 2013; 138–143.
5. Rawassizadeh R, Dobbins C, Akbari M, Pazzani M. Indexing multivariate mobile data through spatio-temporal event detection and clustering. Sensors. 2019 Jan 22; 19(3): 448.
6. Liaqat A, Ilyas S, Mushtaq G. Distributed Computation Offloading of an application from mobile/IoT device to cloud. arXiv preprint arXiv:2302.02481. 2023 Feb 5.
7. Ali MM. Towards Secure Cloud Storage Services. Dissertation. Fargo, USA: North Dakota State University; 2015. Available from: https://www.academia.edu/101762626/Towards_Secure_Cloud_Storage_Services
8. Malik SU, Akram H, Gill SS, Pervaiz H, Malik H. EFFORT: Energy efficient framework for offload communication in mobile cloud computing. Softw: Pract Exp. 2021 Sep; 51(9): 1896–909.
9. Deochake S. Cloud cost optimization: A comprehensive review of strategies and case studies. arXiv preprint arXiv:2307.12479. 2023 Jul 24.
10. Doyle J, Giotsas V, Anam MA, Andreopoulos Y. Cloud instance management and resource prediction for computation-as-a-service platforms. In 2016 IEEE International Conference on Cloud Engineering (IC2E). 2016 Apr 4; 89–98.
11. Henzinger TA, Singh AV, Singh V, Wies T, Zufferey D. A marketplace for cloud resources. In Proceedings of the tenth ACM international conference on Embedded software. 2010 Oct 24; 1–8.
12. Gul OM. Heuristic Resource Reservation Policies for Public Clouds in the IoT Era. Sensors. 2022 Nov 22; 22(23): 9034.
13. De Sensi D, De Matteis T, Taranov K, Di Girolamo S, Rahn T, Hoefler T. Noise in the clouds: Influence of network performance variability on application scalability. Proc ACM Meas Anal Comput Syst. 2022 Dec 8; 6(3): 1–27.
14. Gholami A, Laure E. Security and privacy of sensitive data in cloud computing: a survey of recent developments. arXiv preprint arXiv:1601.01498. 2016 Jan 7.
15. Shi Y. Data Security and Privacy Protection Data Security and Privacy Protection in Public Cloud. arXiv preprint arXiv:1812.05745. 2018 Dec 14.
16. Abdullah L, Freiling F, Quintero J, Benenson Z. Sealed computation: abstract requirements for mechanisms to support trustworthy cloud computing. In International Workshop on Security and Privacy Requirements Engineering. Cham: Springer International Publishing; 2018 Sep 6; 137–152.
17. Frimpong T, Hayfron Acquah JB, Missah YM, Dawson JK, Ayawli BB, Baah P, Sam SA. Securing cloud data using secret key 4 optimization algorithm (SK4OA) with a non-linearity run time trend. PloS one. 2024 Apr 16; 19(4): e0301760.
18. Chari KK, Krishna M. An Efficient Scalable Data Sharing in Cloud Storage Using Key Aggregate Encryption. International Journal of Science Engineering and Advance Technology (IJSEAT). 2015; 3(11): 945–6.
19. Vidhisha G, Surekha C, Rayudu SS, Seshadri U. Preserving privacy for secure and outsourcing for linear programming in cloud computing. arXiv preprint arXiv:1211.1457. 2012 Nov 7.
20. Mohammed B, Kiran M, Maiyama KM, Kamala MM, Awan IU. Failover strategy for fault tolerance in cloud computing environment. Softw: Pract Exp. 2017 Sep; 47(9): 1243–74.
21. Chihoub HE, Ibrahim S, Antoniu G, Perez MS. Harmony: Towards automated self-adaptive consistency in cloud storage. In 2012 IEEE International Conference on Cluster Computing. 2012 Sep 24; 293–301.

22. Alhamazani K, Ranjan R, Jayaraman PP, Mitra K, Liu C, Rabhi F, Georgakopoulos D, Wang L. Cross-layer multi-cloud real-time application QoS monitoring and benchmarking as-a-service framework. IEEE Trans Cloud Comput. 2015 Jun 17; 7(1): 48–61.
23. Scheuner J, Leitner P, Cito J, Gall H. Cloud work bench--infrastructure-as-code based cloud benchmarking. In 2014 IEEE 6th International Conference on Cloud Computing Technology and Science. 2014 Dec 15; 246–253.
24. Jaddoa A, Sakellari G, Panaousis E, Loukas G, Sarigiannidis PG. Dynamic decision support for resource offloading in heterogeneous Internet of Things environments. Simul Model Pract Theory. 2020 May 1; 101: 102019.

