Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 6, June 2021: 1480-1488

Computation Offloading in Fog Computing: Use Cases, Techniques & Issues

Jyoti Yadav [a*], Suman Sangwan [b]

[a*] Research Scholar , CSE Department, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat, Haryana, India.
[b] Professor , CSE Department, Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat, Haryana, India.

***Corresponding author:** jyoti89.cse@gmail.com, suman.cse@dcrustm.org

Abstract

In this era, fog computing is a horizontal, physical, or virtual resource emerging paradigm that extends cloud computing to the edge of the network. Now a day, there are many resources hungry applications i.e. face recognition, augmentation reality, video surveillance, speech recognition, virtual reality that requires high computation capabilities and large memory, which are not available at the end devices. Moreover the distance between the cloud and the end devices may also be an issue for delay sensitive applications. This paper present a comprehensive survey of computation offloading in fog computing. It reviews different techniques used to implement computation offloading along with variation parameters. This study also represents that there are various use cases of computation offloading in real world environment. Finally, open research issues related to computation offloading are also investigated to point out future research directions for the efficient computation offloading in the fog computing environment..

Keywords: Computation offloading, Delay, Energy, Fog Computing, IoT

1. Introduction

Cisco introduced the term fog computing in 2012[1]. Fog computing is complementary to cloud computing for better Quality of Service(QoS) and Quality of Experience(QoE). It is an intelligent layer sitting between cloud and IoT devices i.e. it is a distributed, heterogeneous paradigm that acts as a middleware between cloud and IoT devices and it can be deployed anywhere like on the tower, on university floor, in the shopping mall, on the street light, in-vehicle, in metro stations, etc. A device with computing power, memory and internet connectivity (Router, Gateway, Camera, Switches, Set-top boxes, Hub, Proxy Servers, etc) can be used as a fog node. Here data is accumulated at network edge like the floor of company, trains, airport, shopping complex, etc. After collecting data, it is required to process that data with minimum latency or in few nanoseconds. Here fog node can reduce latency by sitting at the edge of the network.

Due to geographically distributed, heterogeneity of nodes, the major issue in fog computing is unpredictable environment. There are many variations like network condition, number of task, type of devices, and heterogeneity of application. In this paper, we survey the computation offloading techniques and variation parameters addressed by different authors. We also present practical use cases of computation offloading in fog computing. Finally issues related with offloading a task to either fog node or cloud are also investigated to point out future research directions.

In this paper section II will describe computation offloading with respect to fog computing along with 5 w's and 1'h questions associated with the scenario. Section III will cover offloading algorithms with focus on variation parameters and section IV will discuss about various practical use cases of offloading in real world environment. Section V present various techniques used to implement computation offloading. Finally section VI will discuss about research challenges faced in computation offloading.

2. Computation Offloading

Computation offloading[2] is the practice to offload the resource-intensive task to nearby fog or cloud node to overcome the problem of resource limitation and increasing efficiency and reducing latency. Resource demanding task takes longer time in execution at local device because they have limited resources like primary memory, processing power, etc[3].

During computation offloading many questions arise in mind what to offload, where to offload, when to offload, why to offload, who will offload and how to offload[4].

A. What to offload

One of the major problems is to identify what task could be offloaded and what task should be offloaded. The off-loadable task can be identified manually or automatically. Researchers used different application partitioning algorithms based on different approach such as brute force, Greedy heuristic, fuzzy logic to extract offloadable part. In ULOOF[5] framework granularity for offloading is also defined i.e on class basis or method basis. ULOOF works on method level because at class level both offloadable part and non offloadable part exist.

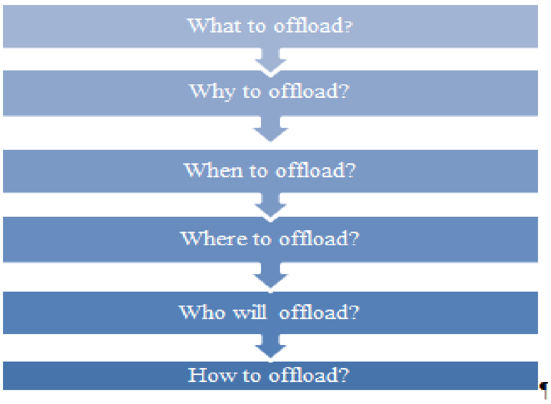

Fig. 1: 5 Q's & 1 H for Offloading

B. Why to offload

To answer this question, there are various criteria based on which decision is taken:

1) Load balancing

It refers to efficiently distributing tasks among backend servers for effective utilization of available resources. For example, a website receiving more than a million requests for image, graphics, audio, video file requires task is to be distributed among different servers for fast processing.

2) Permanent Storage of data

Now a day's Smartphone comes with ample memory space but not good enough to store memory-intensive applications like voice recognition, face recognition, video games, audio files, video files, etc. As a result, the data can be offloaded to higher resource node for permanent storage.

3) Data security

Security refers to preventing data from unauthorized access. A smart phone user can transfer confidential data to the private cloud for security purposes.

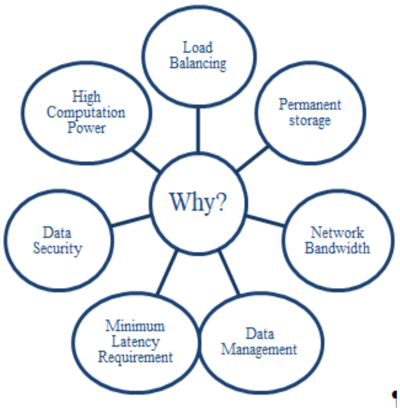

Fig 2: Criteria for why to offload

4) High computation power

If a job is requiring more computing power as compared to its native device, in that case, a complete or part of the job can be offloaded to a remote node or cloud which has high computing capability.

5) Minimum latency requirement

Applications that require real-time processing of data, for example, road traffic data management, patient health monitoring system, etc require minimum latency. If we have limited resources, it will take more time to execute the task locally. If we send a job to the distant cloud, then it will increase latency. In such a situation, a fog node can be a better option for offloading tasks with minimum latency.

6) Data management

In mobile phones, rarely used data is offloaded to fog nodes and whenever that data is required is brought back to the native device. Offloading can be used for organizing and managing data.

7) Network bandwidth

With the emergence of IoT massive amount of data is generated by devices in every millisecond. While limited bandwidth is available, so it's not the right tactic to send all tasks to the distant cloud for execution. Here fog node can be used for pre-processing, analyzing, aggregating data at the edge of the network. So offloading can be used to save network bandwidth[6].

C. When to offload

It is not necessary to always offload the task, because offloading a job adds extra cost, for example, low network speed can increase transmitting time in such a situation some time constraints task should be executed locally. So the decision to when to offload not only depends upon user requirements but also on the current status.

D. Where to Offload

Generally we consider cloud as destination of offloading. There exist various middleware technologies for offloading like cloudlet, micro datacenter, Mobile edge computing, nano data center, famto cloud, etc. Cloudlet is a distributed computing technology that is available at the edge of the network. It is a mini cloud that brings cloud services to the user[7]. Edge computing is the technology that allows computation at the edge of the network[8]. It improves latency, QoE, context-aware service, user experience, and reduce bandwidth consumption[9]. Micro Datacenter is a modular, centralized, and portable data center. Femtocell connects using broadband and DSL Connection to the service provider's network[10]. Where to offload the task either at cloud or middleware technology or execute at native device is to be considered for offloading the task.

E. Who will offload

In fog computing during computation offloading, one of the big tasks is who will act as a decision engine. Some authors prefer gateway as a decision engine others prefer mobile node to work as a decision engine. The decision engine is also called orchestrator and it is responsible for decision of whether the task should be offload or execute it locally. Location of decision engine also play major role in offloading process. Three approaches are used for orchestration centralized, distributed or clustering.

F. How to offload

Various techniques offload tasks at a different level, and we can classify them into system level, application level, and method level. The system-level offloading means to move the entire operating system and its applications on a different node. The application-level technique usually runs a set of apps on the cloud, and the client can call it as web services. The method-level approach partitions a program into pieces and decides which part should run remotely.

The following table represents different variation parameters considered for addressing computation offloading in fog computing.

Variation Parameters

Functionality of offloading system is measured with the help of different variation parameters. These parameters may be in the context of network, availability of resources for execution, or applications. In the list of parameters delay, bandwidth, channel capacity, data rate comes under network context. Energy, computation capability, and memory are considered under device context. Heterogeneity, concurrency comes under application context. Table I shows different variation parameters considered by different authors for offloading algorithm.

Table I. List of variation parameters considered for performance evaluation of computation offloading architecture

Paper	**Parameters**							
	Delay	**Energy**	**Channel capacity**	**Cost**	**Data rate**	**Computation capability**	**Heterogeneity**	**Band width**
[11]	Yes	Yes						
[12]	Yes	Yes		Yes	Yes			
[13]	Yes	Yes						
[14]			Yes			Yes		
[6]	Yes	Yes						
[15]								Yes
[16]	Yes	Yes				Yes		
[17]				Yes			Yes	

4. Use Cases of Computation Offloading

Computation offloading in fog computing has holds promising capabilities for various applications. These applications may be interactive or non-interac tive. This potential has been unveiled through several use cases.

A. User behavior driven health care monitoring

[6] Proposed a technique that involves human-driven, device-driven intelligence as a key component to minimize latency and energy consumption via case studies. The first case study makes use of the machine learning to recognize user behaviors for health care monitoring and perform adaptive low-latency Medium Access Control (MAC)-layer scheduling amid sensor devices. In the second case study, they designed an algorithm for task offloading using which devices can take offloading decisions with minimum latency and energy consumption.

B. Counting number of persons in the image

A prototype of a smart offloading framework designed to work in IoT devices using the Fuzzy Multi-Criteria Decision Making as the decision tool [18],. Network condition is considered for making the decision of whether execution will be done at IoT devices or cloud. A set of the linguistic variable is defined to set offloading criteria. The membership function represents the degree to which the criteria belong to a set of assessment methods or to select the decision method. Hence, a proposed algorithm will work for real IoT devices.

C. Scanning car number to detect car detail

A framework that supports IoT applications with adaptive computation offloading capabilities is designed[19]. Firstly they proposed a design pattern to enable an application to be computation offloaded on demand. After that, an estimation model is employed to automatically decide the deployment plan for offloading. Then the framework is proposed which supports the design pattern and estimation model. Then this framework is used for real-time case study i.e license plate recognition system to collect car details by scanning the number plate.

D. Location-based service

Now a day location-based service had evolved into smart location-based service. These smart location-based services provide many services based on location such as entertainment, traffic, weather condition, etc. So smart location-based service required a complex operation to be executed that's why it requires the task to be offloaded to the fog nodes[20].

E. Game theory

Computation offloading game represents the competition among IoT users and optimized allocation of the processing power of fog nodes[16]. Here resources are allocated hierarchically in fog nodes and clouds. The main aim is to reduce delay and computation energy and maximize the quality of experience and to know offloading decisions for each problem incoming to the IoT user whereas fog nodes work in cooperation to offload task either to neighboring nodes or to the cloud.

Table II. Use cases of computation offloading in fog computing

Paper	Use cases	Details
[6]	User behavior-driven health care monitoring	User behavior-driven health care monitoring application is introduced in the paper
[18]	Counting the number of persons in the image	An algorithm for counting no of objects in an image by using deep learning
[19]	Scanning car number to gather car details	A framework for adaptive computation capability to design pattern and to estimate pattern proposed.
[20]	Location-based service	Running a virtual machine on low-performance network platform degrade quality of service while fog computing and smart virtual machine provide effectively and efficiently location-based service
[16]	Game theory	It uses Nash equilibrium near-optimal resource allocation mechanism
[21]	Environment monitoring system	It utilizes fog architecture for environment monitoring and hazard detection by deploying a computation-intensive applications at sensors.

F. Environment monitoring system

The distributive optimization framework implements cooperative fog computing[21]. Here a problem is divided in N number of tasks and each fog node is capable to execute the task. This subtask optimization is monitored by the work forwarding coordinator. Each fog node executes three cases if it can execute a complete task then it does not pass to other fog nodes or cloud. If it cannot execute tasks then it may pass whole to other nodes or cloud, it may also partially offload workload. So in the proposed algorithm, the unprocessed workload can be shared with the neighboring fog nodes to enhance the quality of experience for the users and power efficiency.

5. Techniques Used For Computation Offloading In Fog Computing

Different authors used various techniques for implementing computation offloading in fog computing in literature. Some of the most popular papers which have used the popular method for performing computation offloading in fog computing are as follows:

A. Energy-efficient optimization for computation offloading in Fog computing system

Energy-efficient optimization for computation offloading solved the problem of energy optimization for a fog computing [22]. Two queuing models one at a mobile device and another at fog node is used to study energy consumption and delay in fog computing. Computing capability and wireless transmission are explicitly considered. The heterogeneity factor is also considered at the network level. Authors formulate an energy-efficient optimization problem that reduces energy utilization by finding transmit power and offloading probability. They have used ADMM based method to address the optimization problem.

B. A new hybrid adaptive GA-PSO computation offloading algorithm for IoT and CPS context application

The hybrid adaptive GA-PSO computation offloading algorithm reduces energy consumption and execution time [23]. Here Particle Swarm Optimization (PSO) and Genetic Algorithm is utilized to find out the solution for scheduling off loadable components in an application. First Task Interaction Graph(TIG) is created for mobile devices where a node represents computational task and edge represents the relationship between nodes. Then the weight is assigned to each node and cost is associated with the edge. The partitioning problem is expressed as a minimization problem.

Table III. Techniques used for computation offloading in fog computing

Technique	Paper
Nash Equilibrium	[16], (Liu, Chang, et al., 2018), [17]
Markov Model	[25]
Fuzzy	[6], [18]
Genetic Algorithm	[23], [26]
Q learning (Reinforcement Learning)	[27]
Multiple Choice Knapsack Problem	[15]
Interior Point-Based Method	[12]
Alternate Direction Method of Multiplier	[22]
Distributed Alternate Direction Method	[21]

C. Socially aware dynamic computation offloading scheme for Fog computing system with energy harvesting devices

Socially aware computation offloading algorithm considered the social relationship of energy harvesting mobile devices [24]. A game-theoretic approach is used to minimize the social group execution cost and to model the interaction among groups. To analyze the delay performance during the offloading process different queuing models are used. Firstly, they utilize the exponential penalty function method to penalize the coupling constraints and convert the inventive Generalized Nash Equilibrium Problem into a classical Nash Equilibrium Problem. Then they synthesized the Karush-Kuhn-Tucker conditions for the smoothing penalized Nash Equilibrium Problem into a system of nonsmooth equations, and then apply the semi-smooth Newton method with Armijo line search to resolve the system.

D. Autonomic computation offloading in mobile edge for IoT applications

Autonomic computation offloading implemented deep Q learning-based autonomic management framework to handle computation resource demand from mobile devices [27]. Here the problem is represented using the Markov decision process (MDP) and the solution is given through reinforcement learning. The randomness of the demand for resources and the mobility of the devices is considered for decision making. This algorithm minimizes the latency of service computing.

E. Multiobjective Optimization for computation offloading in fog computing

Multiobjective Optimization for computation offloading have used queuing theory for balancing energy consumption and delay and estimated payment cost of computation offloading in fog computing [12]. Here three queuing model is used to maintain queue at fog node, mobile node and cloud, data rate and power consumption of wireless link are explicitly considered. Authors formulated a multi-objective optimization problem to minimize execution delay, payment cost, and energy utilization by finding the best offloading probability and transmission power of the mobile device. Using the scalarization method they can change the multi-objective problem into a single-objective problem. Interior Point Method (IPM) is used to address the transformed optimization problem.

F. Computation offloading and resource allocation in mixed Fog/Cloud computing systems with Min-Max fairness guarantee”

In this paper[13], the authors proposed Offloading Decision Making and Resource Allocation Algorithm (CORA) algorithm where semi definite relaxation (SDR) and random extracting are used for offloading decision. Bisection Method for Computation Resource Allocation Algorithm (BCRA) algorithm was proposed to solve computation resource allocation in order to solve the nested resource allocation problem in CORA.

G. Scalable fog computing with service offloading in bus networks

In this paper [26], the authors considered the scalable fog computing paradigm with service offloading in the bus network. The main point is that not only mobile devices can use the fog server services but also overloaded roadside cloudlet can offload the computational task to the fog server. So it also helps in enhancing the capabilities of the cloudlet also. They have used a genetic algorithm for allocation purposes and improve user experience also. Scalable fog computing paradigm used a cyclic approach. The first estimation of the volume of a computational task is made by roadside cloudlets. Then it does planning for task allocation. After that using the access point task is offloaded to the fog servers. This minimizes the cost of transmitting data and enhances user experience.

VI. Open Issues And Future Directions

In this section authors recapitulate the challenges and open issues associated with computation offloading in fog computing. Authors have noticed that the offloading algorithm generally focuses on one part of computation offloading. It may be in reference to either what, why who, how or when. Mostly algorithm works on few parameters and improves them. But there should be a combined decision based on what, who, why, how and when the offloading should be done.

The following are the major issues associated with computation offloading in fog computing.

A. Platform Variety

One of the challenges is variation and heterogeneity in end devices and fog node used in computation offloading. A standard offloading framework is still a challenging issue.

B. Fault Tolerance

In fog computing environment end devices can move from one location to another then network bandwidth and data exchange rate may vary or connection may be lost. In such cases offloading algorithms should be provided with suitable fault tolerance mechanisms.

C. Security

There may be some security and privacy issues associated with computation offloading. When we send some critical and confidential tasks to other fog nodes for computation purpose then the risk may arise. So the computation offloading algorithm must follow some security mechanism while offloading a task to distant fog node.

D. Context Awareness

Offloading decisions must be on the basis of different offloading parameters. Offloading will not be yielding positive results all the time. So the offloading decision must be based upon the current context of the situation. Here context will refer to end device resources, network condition and fog node resources.

E. Service Level agreement

Offloading a computing task involves end device, fog node and cloud node. There should be some mechanism which ensures that no service level agreement violation during offloading.

VII. Conclusion

Fog computing allows shifting of resource-intensive tasks to the edge of the network by using computation offloading. It reduces energy consumption, latency and improves QoS and user experience. This paper has identified major components associated with computation offloading in fog computing which includes what, where, why, how and who will offload task to fog node. This paper is focusing on the use cases of computation offloading and the techniques used for the implementation of computation offloading in fog computing. Authors also identified variation parameters which were used by different authors for the evaluation of algorithms. There were several architectures proposed for fog computing but they were theoretical because fog computing is in the initial stage. With the emergence of IoT computation offloading gets momentum in research.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” *MCC’12 - Proc. 1st ACM Mob. Cloud Comput. Work.*, no. March, pp. 13–15, 2012.

[2] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” *Mob. Networks Appl.*, vol. 18, no. 1, pp. 129–140, 2013.

[3] S. Deshmukh and R. Shah, “Computation offloading frameworks in mobile cloud computing : A survey,” *2016 IEEE Int. Conf. Curr. Trends Adv. Comput. ICCTAC 2016*, 2016.

[4] H. H. Elazhary and S. F. Sabbeh, "The W5Framework for Computation Offloading in the Internet of Things," *IEEE Access*, vol. 6, no. c, pp. 23883–23895, 2018.

[5] S.-Y. Yu, S. Secci, D. F. Macedo, R. Langar, J. L. D. Neto, and J. M. S. Nogueira, "ULOOF: A User Level Online Offloading Framework for Mobile Edge Computing," *IEEE Trans. Mob. Comput.*, vol. 17, no. 11, pp. 2660–2674, 2018.

[6] Q. Duy, M. V Ngo, T. Quang, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digit. Commun. Networks*, vol. 5, no. 1, pp. 3–9, 2019.

[7] F. V. Vargas, "Cloudlet for the Internet-of- Things," 2016.

[8] W. Shi, J. Cao, S. Member, Q. Zhang, and S. Member, "Edge Computing : Vision and Challenges," vol. 3, no. 5, pp. 637–646, 2016.

[9] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 278–289, 2018.

[10] A. Mukherjee and D. De, "Engineering Science and Technology , an International Journal Low power offloading strategy for femto-cloud mobile network," *Eng. Sci. Technol. an Int. J.*, vol. 19, no. 1, pp. 260–270, 2016.

[11] X. Meng, W. Wang, and Z. Zhang, "Delay-Constrained Hybrid Computation Offloading with Cloud and Fog Computing," *IEEE Access*, vol. 5, no. c, pp. 21355–21367, 2017.

[12] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective Optimization for Computation Offloading in Fog Computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, 2018.

[13] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog / Cloud Computing Systems with Min-Max Fairness Guarantee," vol. 6778, no. c, 2017.

[14] F. Chiti and R. Fantacci, "A Matching Theory Framework for Tasks Offloading in Fog Computing for IoT Systems," *IEEE Internet Things J.*, vol. PP, no. c, p. 1, 2018.

[15] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, and D. Soudris, "Computation Offloading and Resource Allocation for Low-power IoT Edge Devices," pp. 7–12, 2016.

[16] H. Shah-mansouri and V. W. S. Wong, "Hierarchical Fog-Cloud Computing for IoT Systems : A Computation Offloading Game," vol. 4662, no. 99, pp. 1–12, 2018.

[17] Q. Li, J. Zhao, Y. Gong, and Q. Zhang, "Energy-Efficient Computation Offloading and Resource Allocation in Fog Computing for Internet of Everything," *China Commun.*, vol. 16, no. March, pp. 32–41, 2019.

[18] W. Wibisono, M. Widhi, P. Putu, T. Ahmad, and R. Anggoro, "An Adaptive Offloading Framework for Improving Performance of Applications in IoT Devices Using Fuzzy Multi Criteria Decision Making," vol. 7, pp. 31–36, 2018.

[19] S. Chen, B. Liu, X. Chen, Y. Zhang, and G. Huang, "Framework for Adaptive Computation Offloading in IoT Applications," pp. 2–7, 2017.

[20] Y. Son and Y. Lee, "Offloading Method for Efficient Use of Local Computational Resources in Mobile Location-Based Services Using Clouds," *Mob. Inf. Syst.*, vol. 2017, pp. 1–9, 2017.

[21] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," *Proc. - IEEE INFOCOM*, 2017.

[22] Z. Chang, T. Ristaniemi, S. Mao, X. Guo, and L. Liu, "Energy Efficient Optimization for Computation Offloading in Fog Computing System," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, 2017.

[23] R. Ezhilarasie, M. S. Reddy, and A. Umamakeswari, "A New Hybrid Adaptive GA-PSO Computation Offloading Algorithm for IoT and CPS Context Appplication," 2019.

[24] L. Liu, Z. Chang, S. Member, and X. Guo, "Socially-aware Dynamic Computation Offloading Scheme for Fog Computing System with Energy Harvesting Devices," vol. 4662, no. c, pp. 1–13, 2018.

[25] [25] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, “On Reducing IoT Service Delay via Fog Offloading,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, 2018.

[26] [26] D. Ye and M. Wu, “Scalable Fog Computing with Service Offloading in Bus Networks,” 2016.

[27] [27] M. G. R. Alam, M. M. Hassan, M. Zi. Uddin, A. Almogren, and G. Fortino, “Autonomic computation offloading in mobile edge for IoT applications,” *Futur. Gener. Comput. Syst.*, vol. 90, pp. 149–157, 2019.