ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

Development of an improved method for the placement of IOT applications in a fog computing environment based on the particle swarm algorithm

Reyhaneh Novin [1] ,Mohammad Faraji Mehmandar*[2]
Department of computer engineering, Parand Islamic Azad University, Tehran, Iran

Abstract:
Internet of Things (IOT) has made the connection and communication among the present objects possible which has led to the unprecedented production of large and heterogeneous data which is known as data explosion. Cloud computing is commonly used to organize these data. The challenges such as increasing real-time requests or applications which are sensitive to latency and limited network bandwidth cannot be solved using cloud computing; fog computing was proposed as a supplement to cloud computing solution to solve this cloud computing problem. Fog computing extends the cloud services to the edge of network, providing communication and storage close to edge devices and end users. The goal is to improve in reducing latency, mobility, network bandwidth, security and privacy.
To take full advantage of fog computing, a great deal of research is needed on a wide range of topics. One of the most important issues in fog computing is the optimal and efficient placement of IoT applications which optimize one or more objectives. This paper presents an improved method for the placement of IoT applications in a fog computing environment due to the particle swarm algorithm that "proposed" approach was analyzed with the layout approach in the cloud computing presented in experiment 1. Based on this, service latency performance criteria, the number of allocated containers, and total cost were compared. The test results show that the proposed approach has improved the service latency parameter up to 40%, the total cost up to 23%, and the allocated containers up to 12%.
In Experiment 2, the proposed solution was compared with other layout methods in fog computing to confirm its effectiveness. The test results show that the average latency in the proposed algorithm is 3.1% and 6.2% less than other methods, respectively, leading to cost minimization of 5.7% and 15%, respectively.
Keywords: Fog Computing, Cloud Computing, Edge Computing, IoT, Applications, Application placement, Optimization Algorithms[1]

1. Introduction:

The Internet of Things is a new concept in the world of technology and communication. In fact, the Internet of Things is a modern technology in which intelligent objects around us interact and collaborate to achieve common objectives. Just as the Internet connects all people, so the Internet of Things connects all things together and can send data through communication networks.
Every online object, such as smart cameras, wearable sensors, environmental sensors, smart home appliances, vehicles, etc., are the IOT components. However, the Internet of Things enhances the quality of human life, but it should be borne in mind that using the Internet of Things generates large amounts of data, these data require storage and analysis systems. [1-3]
Typically, [4,5] cloud computing is an environment to store and process data generated by the Internet of Things. Access to information is possible anywhere and anytime with cloud computing. Cloud computing has several advantages such as cost reduction, easy access to information, increased storage, flexibility, efficient resource allocation, and reduced execution time, but cloud computing also has disadvantages. Some of these problems are caused by the benefits it provides. For instance, the placement of data processing centers on the Internet platform does not allow the user to be able to process data regardless of location and, this means that the user faces unpredictable latencies and various security issues. Because the data will be entrusted to a third party. As a result, transferring large volumes of data to the cloud for storage and processing causes the network bandwidth's saturation... While, many applications, such as health monitoring programs, require low latency [6]. Latency caused

[1] Department of computer engineering, Parand Islamic Azad University, Tehran, Iran
[2] * Corresponding Author: Mohammad Faraji-Mehmandar
Department of computer engineering, Parand Islamic Azad University, Tehran, Iran
E-mail: m.Farajim@iau-tnb.ac.ir

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

by data transfer to the cloud and then to the software seriously affects the performance of these applications. A new concept called cloud computing was used to address these problems [4,7,8]. The fog computing features include support for applications with low latency, location awareness, and real-time interaction. Fog computing is a new term in the world of computing called extended cloud computing. Fog computing is not a substitute for cloud computing, it solves many cloud computing problems and makes it more efficient. One of the major challenges in fog computing is the placement of IOT application services [9]. Since the existence of hardware and software resources in the fog network environment to respond to user requests require solutions and processes to release them, resource management is always the focus of engineers and researchers in this field. Considering the placement of service is one of the difficult problems and cannot be solved in polynomial time, so it can be discussed in the form of an optimization problem; in such a way that with proper placement of the services, system efficiency has increased, and response time is reduced. Resource placement plays an important role in system performance and increasing the level of customer satisfaction. In any optimal resource placement strategy, factors such as minimizing the response time and latency and maximizing the operational capacity should be considered [8,10,11]. We intend to provide an improved approach to service placement in the field of fog computing due to the particle swarm optimization algorithm to efficiently place services on fog sources to reduce the cost and response time. The particle swarm optimization algorithm is an efficient optimizer which has proven its ability in various optimization problems. Moreover, we use the Java simulation environment to simulate resource management and resource allocation policies.

Other goals that we seek to achieve in this study are included:

- Study and classification of resource management methods available in cloud and fog networks
- Providing an improved way to deploy IOT applications in the fog computing environment due to particle swarm algorithm
- Service placement to reduce the cost and response time of applications
- Placement of the services on the fog resources instead of cloud resources

The second section deals with the research literature which includes research related to the placement of resources and their study and the classification of techniques and algorithms in this field. The third section provides the required background on resource placement issues in fog computing. In the fourth section, we propose a method that explains the implementation of the proposed algorithm in the system models. In the fifth section, we explain the simulation results, and in the sixth section, including conclusions and finally, recommendations are provided for future work in this field.

2. Review of the literature and research background

This part indicates several reviewed articles on resource management problems on fog and edge calculations.

Skarlat et al. [12] discussed the service placement for IOT environment in fog calculation to regard QOS resources and criteria' heterogeneity. They proposed a genetic algorithm to find a proper solution for placement of the service in the fog landscape and indicating, through experimental results, that their solution was more effective than applying fog sources and the execution time of IOT applications. They also explained a conceptual framework for the issue of service placement in the Fog landscape.

Mohammad Faraji Mehmandar et al. [13] this paper first presents a distributed computational framework for automated resource management in computing, then a customized version of the IOT service system, a system of reinforcement learning as a decision maker in the planning stage, and a vector regression method. Uses the analysis stage. At the end of a series of simulation experiments to evaluate the performance of the system, the results show that the average latency, cost and latency have been reduced by 1.95, 11 and 5.1%, respectively, compared to the existing solutions.

Mahmoud et al. [14] suggested quality of application placement approach from the experience due to fuzzy logic models to prioritize multiple requests of presenting the utilizations to simplify using applications in fog sources and categorize fog sources. Besides, a linear optimization problem was presented to ensure the quality of the user experience.

Velasquez et al. [15] have designed a modular architecture to identify IOT Internet services. The suggested architecture aims to prepare an information service delivery system which facilitates IOT service placement in a convenient location due to specific requirements the proposed architecture contains three major modules: service repository, data collection, and service orchestrator. As the core of the architecture, the service orchestrator module is responsible for implementing the policies that make decisions about service placemen. They also formulated the task of service placement utilizing the correct linear programming to achieve architectural optimization objectives.

Mahmoud et al. [16] proposed a quality of program location approach from experience based on fuzzy logic models to prioritize different requests. Provide applications to simplify the application of applications in foggy sources and

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

categorize foggy sources. In addition, a linear optimization problem is presented to ensure the quality of the user experience.

Naranjo et al. [17] designed a new framework in the fog environment to manage smart urban devices to meet scalability, latency requirements, and energy consumption. Besides, three kinds of communication among components for traffic management and tasks at fog nodes are explained: Communication between primary, first and secondary. Experimental results indicate that the suggested framework is suitable to achieve smart and intelligent devices in fog environments.

Yao et al. [18] investigated the placement of cloudlet servers with multiple capacities of the resource, and the user-cloudlet was examined in a cost-effective way without the certain violation. The suggested solution is divided into two parts: the issue of selecting a heterogeneous server and the issue of placement. For the sub-problem of selecting a cloudlet server, a greedy algorithm with the least cost was presented, and both the resource capabilities and the user mobility topology were regarded for the placement problem. The issue of cloudlet placement has also been formulated as a proper linear programming approach which should be solved during polynomial complexity.

Masoumeh Etemadi et al. [19] In this paper, an efficient resource approach is presented. In this method, we use an autonomous computational model to decide on increasing and decreasing the dynamic scale of fog. We also provide an independent resource framework based on the three-layer fog architecture. We design the harness and finally test the effectiveness of our solution with three workload tracks. The simulation results show that the proposed solution reduces the total cost and latency violation.

Yousefpour et al. [20] have introduced the issue of preparing dynamic fog services which dynamically applies IOT services on fog sources to meet QOS requirements. The proposed issue is to try to achieve a better QOS for the users to reduce the service latency and use bandwidth utilization while minimizing the use of cloud supply resources. They formulated the suggested issue in the form of correct linear programming and suggested two exploratory approaches: Min-viol heuristic to minimize the SLA and Min-heuristic costs to decrease total resource costs. Besides, the exploratory approaches are assessed applying real-world traffic routes which indicates the usefulness of suggested solution compared to other exploratory methods.

Brogi and Forti [21] have suggested a QOS awareness-raising method for multi-component IOT utilizations on fog infrastructure. The suggested method can find eligible developments of Internet applications on a fog infrastructure in a context-aware approach. Furthermore, they sampled a Java-based tool called Fog Torch which handles the suggested method to confirm their technical feasibility.

The authors of this paper [22] have analyzed the suggested design of IOT applications, their related frameworks, and best practices for distributed systems to design and implement a dynamic, flexible fog computing framework. Besides, they facilitate the communication among devices, orchestration of fog devices, placement of IOT services, and provision of dynamic resources in the fog environment.

The authors of this book [23] seek to reassure us that IOT devices themselves can enhance the security of networks they use. Now, the network itself has very limited knowledge about the type of data which should be sent and what should not be. Devices can be hijacked to attack other devices or the network itself. By reviewing this book, we can get closer to solving problems such as IOT challenges, scalability, and interoperability. This book helps us get there. It is an educational resource that illustrates the principles of IOT coherently and comprehensively. IOT is ready to change our world, and this book provides the foundation for understanding and scanning the changing IOT environment.

In this paper [24], a proposed scheme for data analysis due to fog computing with cost-effective resources can be used to provide resources for fog-based IOT applications with a resource cost optimization approach for IOT crowd sensing purposes, and a new platform for smart city applications is examined. MIST scheme is capable of a rapid response at three different levels (such as edge, fog, and cloud level), providing high computing performance in the smart cities of the future. To address the welfare cost problems in this scheme, the consumer data and virtual machines (VMs) were jointly examined; finally, a comparative performance evaluation of cloud computing with fog computing was performed for a city or a large number of Internet-connected devices (e.g., Tehran province).

The authors of this paper [25] review the advances in cloud computing architecture and provide guidance for additional research. Articles published in journals, conferences, white papers were reviewed. This work aims to identify, review and explain current trends and advances in cloud computing architecture. However, only 13% of articles reviewed discussed the services of others as a service, while only 26% of articles reviewed the covered topics related to the main actors involved in cloud computing. It will identically benefit cloud providers, users and researchers.

The design proposed by the authors [26] introduced fog and edge computing as a model in which computing power moves to the sources where the data is generated. The main goal is to accurately understand the concept of fog and edge computing and the opportunities offered to consumers and organizations.

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

This paper [27] proposed a design for an edge node resource management framework to enable fog computing and to address resource management problems to prepare edge nodes for cloud computing, establishing workloads on the edge nodes, and allocating dynamic resources at the edge nodes. For this purpose, they have proposed a supplier mechanism for accessing services, deploying workloads, and terminating edge services. It creates an automated scalability mechanism for network edge resources' dynamic management based on a linear search algorithm
.

Table 1-2. Work done on resource management in May

Reference	**Applied method**	**Auto-scaling Policy**	Performance evaluation criteria	Simulation tool	Advantage	Weakness
[12]	Meta-Heuristic-based (GA)	Proactive	Latency, cost	(IFogSim) Simulation	Comprehensive and complete formula	Low scalability
[13]	Learning-Based (*RL*)	Proactive	Latency, Total Cost Energy	Java Program	Low computational complexity	Low scalability
[14]	Heuristic based	Reactive	Placement time Deadline Number of fog nodes	(IFogSim) Simulation	Reduce time	Failure to evaluate the proposed policy in a case study
[15]	Meta-Heuristic-based (Lip)	Proactive	Latency	matlab	Reduce service latency	Lack of evaluation of energy consumption, Lack of a proper simulation
[16]	Learning-Based(Fuzzy logic)	Proactive	application placement time, processing time	(IFogSim) Simulation	Reduce processing time	Failure to evaluate the proposed algorithm in a real case study, Failure to review the results of the approach
[17]	Heuristic based	Reactive	energy	(IFogSim) Simulation	Low energy .1	Low scalability .1
[18]	Heuristic based	Reactive	cost	(IFogSim) Simulation	Low computational complexity	Failure to evaluate the proposed algorithm in a real case study
[19]	Learning-Based (*HMM*)	Proactive	Latency Total Cost	IFogSim Toolkit	Low computational complexity, Reduce latency	Failure to evaluate the proposed algorithm
[20]	Heuristic		Service latency,	Simulation	Delay ,	Lack of evaluation of

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

	based(*Greedy Algorithms*)	Reactive	Number of fog. Services, costs	(Java)	Low computational complexity	energy consumption
[21]	Heuristic based (Backtracking)	Reactive	Design time, Placement time, execution time	matlab	Reduce latency, Bandwidth	High computational complexity, Lack of a proper simulation, Lack of evaluation of energy consumption
[22]	Heuristic based(Vm docker)	Reactive	Service input patterns	simulation	reduction in costs, Check the deployment time	Error tolerance mechanism, Reset the device to a hazy environment
[23]	Model Based	Proactive	Fog nodes	simulation.	Proper design and construction of IoT networks	Security, Low information confidentiality
[24]	Heuristic-based (MINLP)	Reactive	Fog nodes	Mixed Integer Linear Programming (MILP)	Minimize costs	Lack of sensing modules in fog layers
[25]	Meta-Heuristic Based	Proactive	User requests	simulation	Ability to access resources anywhere and anytime	
[26]	Model Based	Proactive	Fog and edge nodes	simulation	Location awareness, Low delay	Scalability, Provide dynamic resources
[27]	Model Based(Linear search algorithm)	Proactive	Memories	Apache JMeter11	Minimal reduction, Delays, communication Reduce data traffic, Reduce frequency communication	Lack of framework for single and multi-edge node environments

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

Proposed method	Meta-Heuristic-based (PSO)	Reactive	Cost of Fog node, execution time	Java simulator	Minimize costs, Reduce program run time	

In some studies[12,15,17,19,25], heuristic and metaheuristic algorithms were proposed to monitor some challenges and various requests for the optimal and efficient placement of IOT applications.
Some studies [16,27] were proposed to provide fog services for using specific applications which are conceptually similar to the proposed method, but have different objectives than meeting the low latency or low cost of IOT applications.
Some studies due to axis approximation [20,22] also pursue objectives such as reducing latency and bandwidth to support the entire program cycle, including design time, placement time, and so on.
Although above research have made considerable contributions to us, they cannot be directly used to prepare optimal and effective placement of latency-sensitive IOT applications. The suggested approach is due to a meta-heuristic algorithm and considers the latency and resource cost requirements of very low fog nodes for IOT applications. Finally, the suggested approach utilizes the particle swarm algorithm which is a meta-heuristic algorithm, to decrease the execution time and cost of placement of fog applications. Each of the research discussed has regarded some of the above characteristics, while the suggested approach regards them all together, it fills the gap in the thematic literature on dynamic frameworks for latency-sensitive IOT applications.
This section has examined the challenges of fog computing environment; finally, the particle swarm algorithm required to build the optimal approach is surveyed, then the proposed method due to the particle swarm optimization algorithm is defined, and the fit function for our suggested model due to the defined formula is presented.

3. Challenges of fog computing

There are advantages for not focusing and getting the cloud closer to the edge of the network; however, this new architecture has brought several new issues. Most of the fog computing architecture challenges are due to the decentralized and distributed nature of this structure. These challenges include interoperability infrastructure, monitoring, accountability, virtualization, VM lifecycle, context awareness, resources and tasks, resource location, task scheduling, offloading, usability programmability, session management, and security. Two very important challenges in this structure are resource management and placement [17].
1) Resource placement and management: Resources in fog computing were heterogeneous and distributed which are located in different places, and may belong to different providers. Therefore, special mechanisms are needed to coordinate and monitor these resources. In this dissertation, we seek the optimization algorithm for the optimal placement of resources among fog nodes to maximize the use of network resources and optimize energy consumption.

3.1. Resource Management

Fog computing basically includes the available IOT components which run in cloud data centers, i.e., on switches, routers, proxy servers, receivers, smart ports, base stations, and other fog equipment [1, 6] which provides location awareness, mobility support, context awareness, distributed data analysis, real-time interaction, interface, and mediator heterogeneity, scalability, and interoperability to meet the requirements of latency-sensitive applications. On the other hand, due to the heterogeneity of resources and dynamic negotiations, the highly variable and unpredictable nature of the fog network, it is necessary to consider resource management as one of the challenging issues to increase the efficiency of fog computing.

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

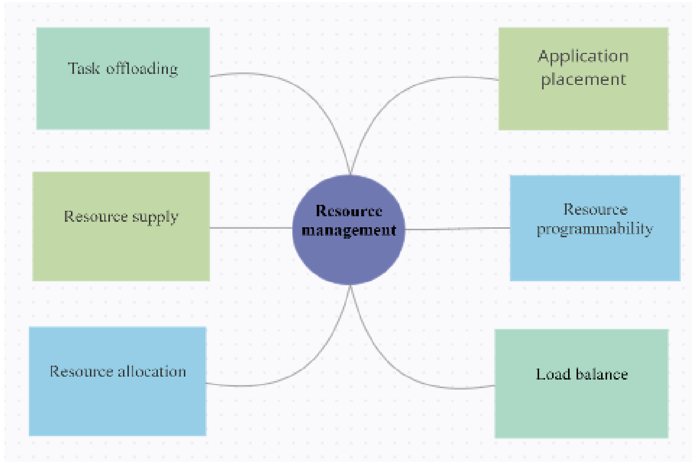

Figure 3-1 .Sub-branches of resource management in May Computing

3.2. Application placement

An IOT application consists of several IOT services extending into virtual fog nodes and interacts with each other to provide proper functionality [9]. The problem of fog service placement in fog landscape is the optimal placement plan between IOT services and fog nodes to maximize the exploitation of fog resources, which at the same time determines the fulfillment of QOS requirements of IOT services. Figure 3-2 shows an example of the placement of an application in the fog computing environment.

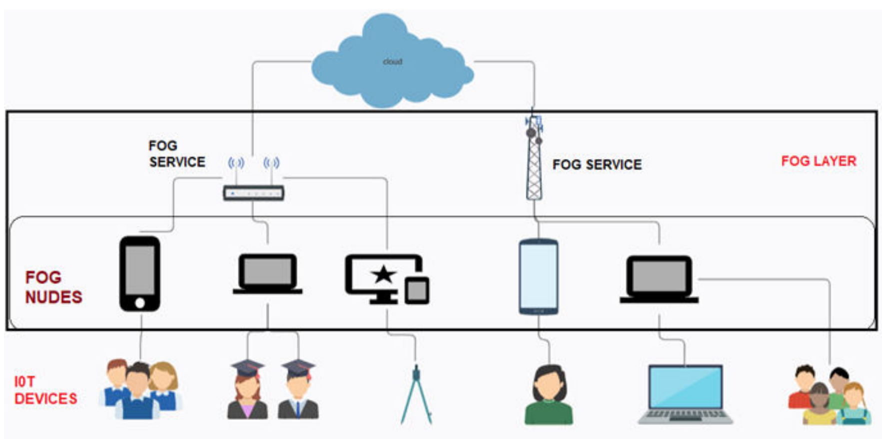

Figure 3-2. The problem of the placement of application in the computer fog

3.3. Particle Swarm Optimization Algorithm

PSO algorithm is a social search algorithm modeled based on the social behavior of flocks of birds. This algorithm was used to discover the patterns governing the simultaneous flight of birds and their sudden change of direction and optimal deformation of their flocks.

Collective intelligence: It is a systematic property in which particles collaborate locally, and the collective behavior of all these particles causes convergence at a point close to the optimal global solution. The strength of this algorithm is the lack of need for global control. Each particle in this algorithm has a relative autonomy which can move across the solution space and collaborate. PSO algorithm is one of the most popular collective intelligence algorithms used to train neural networks. This algorithm was first proposed by Kennedy and Eberhart and was named PSO (Particle Swarm Optimization). Because it was inspired by the group behavior of birds during their flight. The PSO algorithm, like other demographic algorithms, uses a set of possible solutions. These solutions will continue to move until an optimal solution is found or the conditions for the end of the algorithm are met.

$$V_i^{k+1} = WV_i^k + C_1 rand_1 * (pbest_i - X_i^k) + C_2 rand_2 * (gbest_i - X_i^k) \quad (1)$$

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

$X_{i+1}^{k}=X_i^k+V_i^{k+1}$ (2)

Where V_i^{k+1} represents the velocity of particle i in the iteration of k+1, v_i^k is equal to the velocity of particle i in the iteration of k. ω denotes the weight of inertia, c_1 and c_2 are accelerators, $rand_1$ and $rand_2$ are the random numbers between 0 and 1. X_i^k Is the position of the i particle in the k-th iteration. $pbest_i$ is the best particle position of $gbest_i$,i is the position of the best particle in a population. X_i^{k+1} Indicates the position of the particle i in the iteration k+1.

4. The proposed method

Earlier, fog computing was introduced as a set of fog nodes with different types of heterogeneous resources distributed between the user layer and the cloud located on the edge of network. Therefore, these resources' placement plays an important role in the fog computing environment and is considered a challenge in this field. The purpose of the layout of applications in fog computing is the correct and optimal placement of the IOT applications in fog computing environment. Considering the problems presented in the previous sections on placement, applications in the fog nodes are an np-hard problem. A method was presented for the optimal and efficient placement in the fog computing environment and, the response time and cost in the placement of applications were reduced using the particle swarm algorithm. Nodes in sensor networks receive data from their surrounding environment. After receiving the data, the nodes are sent to the gateways and then to the fog devices which are stored and processed in the fog or sent to the cloud. Each application consists of several modules. When the applications start executing on the fog devices, several modules should be run, which requires executing a virtual machine on the fog device. Optimal algorithms are used to obtain the best allocation of virtual machines to modules. The fit function for our proposed model is defined based on Formula 3-1.

4.1. Evaluation function:

The purpose of evaluation function in this section is to provide an improved method for the placement of the IOT applications in a fog computing environment due to a particle swarm algorithm that can reduce latency and costs. For this purpose, we first define the applications of services consisting of each of these services with different functions. The following formula represents the applications, and si represents the services, and P (An) denotes the application execution time. This application execution time is obtained from the difference between the application deadline time and the application response time. R_1 and R_2 are each the response time and cost impact factors. The computation cost in this section also includes the cost of resources and the penalty cost which include the cost of CPU, storage and memory. At present, using the optimization algorithm, a new mechanism is presented to eliminate the latency and maintaining bandwidth which places the applications with the best available node. This will keep the bandwidth constant through which the communication is best established and costs are reduced, and the response time will be minimal.

$$Min(R_1(\sum_{A_n}^{A}(p(A_n).\sum_{S_i}^{A_n}(\sum_{f_i}^{Res\ Si(f)}.x_{si}^{fi}))+(R_2(\sum C_{fi}^{c}))) \quad (3)$$

The computation cost is obtained from the total cost of resources in fog with the penalty cost

$$c_{fi}^{c}=R_{fi}^{c}+P_{fi}^{c} \quad (4)$$

Resource cost in fog are obtained from the sum of CPU processing costs, memory processing costs and storage processing cost.

$$Fog\ resource\ cost=Cost_{fn}^{cpu}+Cost_{fn}^{memory}+Cost_{fn}^{storage} \quad (5)$$

$$Cost_{fn}^{cpu}=\sum_{f\in FN}\sum_{(req\{u,r\}}C_f^p.U_{(req\{u,r\}}^{p}.X_{req\{u,r\}}^{f} \quad (6)$$

$$Cost_{fn}^{memory}=\sum_{f\in FN}\sum_{(req\{u,r\}}C_f^m.U_{(req\{u,r\}}^{m}.X_{req\{u,r\}}^{f} \quad (7)$$

$$Cost_{fn}^{storage}=\sum_{f\in FN}\sum_{(req\{u,r\}}C_f^s.U_{(req\{u,r\}}^{s}.X_{req\{u,r\}}^{f} \quad (8)$$

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

P (An) represents the execution time of program which is the difference between the execution time of application deadline and the response time of application. R_1 and R_2 are each impact factor of response time and cost.

Conditions:
P(An)=D(An)-R(An) [9]

$$\begin{cases} if\ D(An)-R(An) \geq 0 \Longrightarrow D(An)-R(An) \\ else\ D(An)-R(An) < 0 \Longrightarrow 0 \end{cases}$$

$R_1+R_2=1 \qquad R_1>R_2$

symbols	Description	Relevant symbol
A	Application (IOT applications)	execution time
A_k	Application (IOT applications)	execution time
S_i	Service in an application (Services of the IOT applications)	execution time
$P(A_n)$	Application execution time	execution time
$Res_{si(F)}$	All fog cells able to host a service Si.	execution time
f j	Fog cell (node in the fog environment)	execution time
C_{fi}^{c}	Computation cost	Average Cost
R_{fi}^{c}	Resource cost	Average Cost
P_{fi}^{c}	Penalty cost	Average Cost
C_f^p	Cost of process cpu	Average Cost
c_f^m	Cost of process memory	Average Cost
C_f^s	Cost of process storage	Average Cost
U_p^r	The cpu required for each request	Average Cost
U_m^r	The memory required for each request	Average Cost
U_s^r	The storage required for each request	Average Cost
D(A_n)	Application deadline	execution time
R(A_n)	Response time of an application	execution time

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

R_1	Impact factor of the response time on the problem	execution time
R_2	Cost impact factor in the problem	execution time

This study aims to optimize the quality of services provided to customers that the resources proposed by the algorithm cause the sources of fog to be virtualized and reduce the response time of applications.

5. Analysis of the results

The details of results are presented and analyzed in this part. The performance of suggested approach is compared to other placement strategies. To fully understand the advantages of suggested model, we investigated two experiments to survey the impact of various parameters and framework factors influencing the results. Experiment 1 is a survey of the advantages of our proposed algorithm with a cloud placement solution. In Experiment 2, we compare our solution with two new basic approaches, Fog Plan and Optimized.

5.1. Experiment 1:

In this experiment, to validate the effectiveness of proposed solution, a traffic tracking based on the Poisson distribution called Bursty Workload was used to model the requested arrival over a one-day period. In this experiment, time intervals are 5 minutes, the simulation time is 4 hours with 48 intervals, and topology settings include three cloud servers, seven fog nodes, and 30 services. The rate of sudden input traffic of the fog nodes was shown in Figure 1.

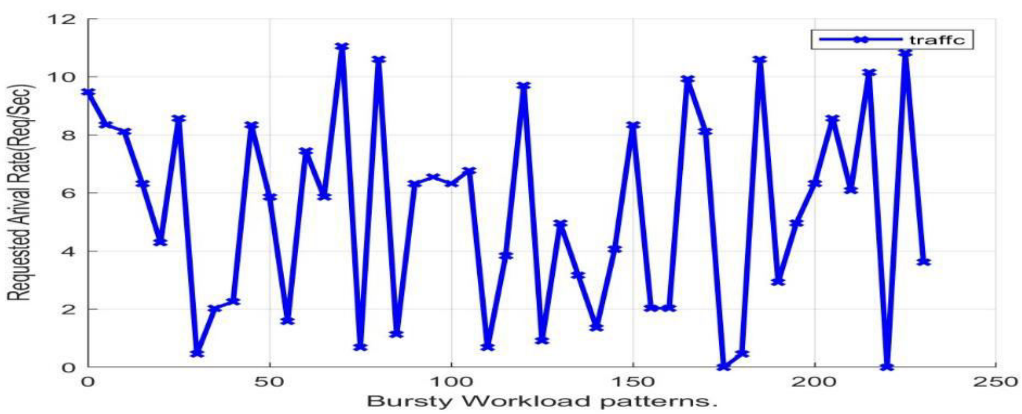

Figure 5-1: Traffic rate

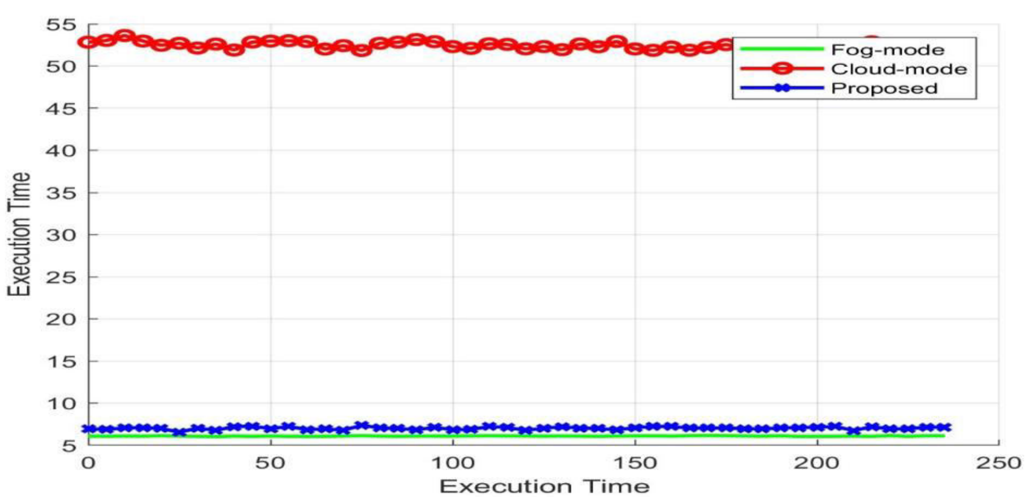

Figure 5-2: Average execution time

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

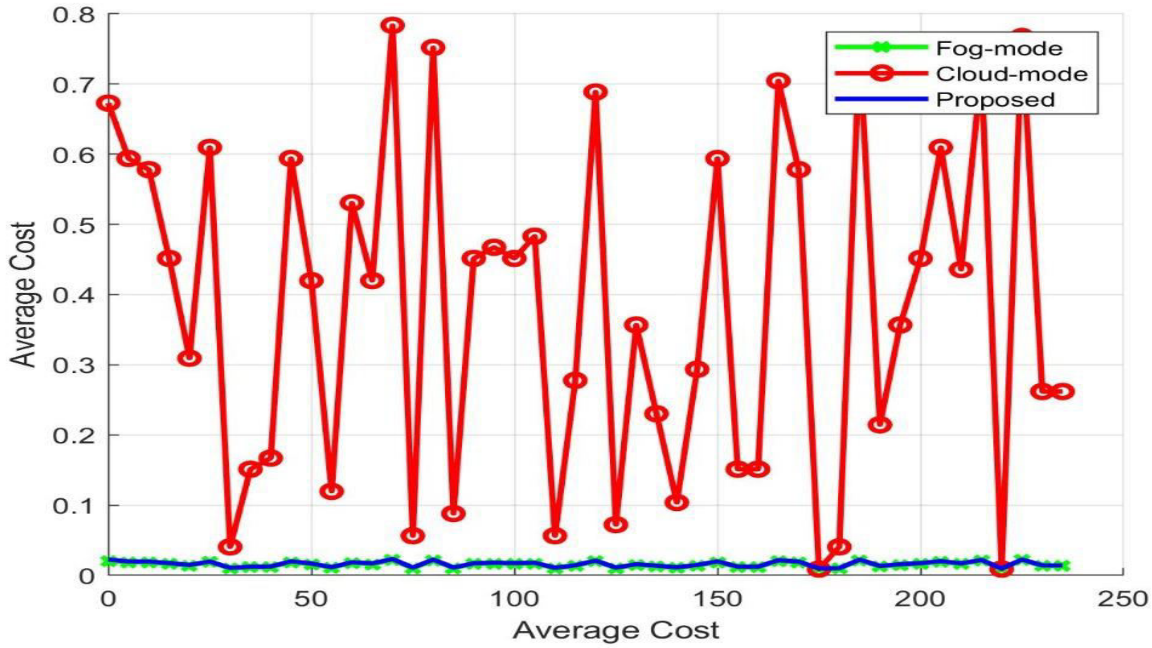

Figure 5-3: Average Cost

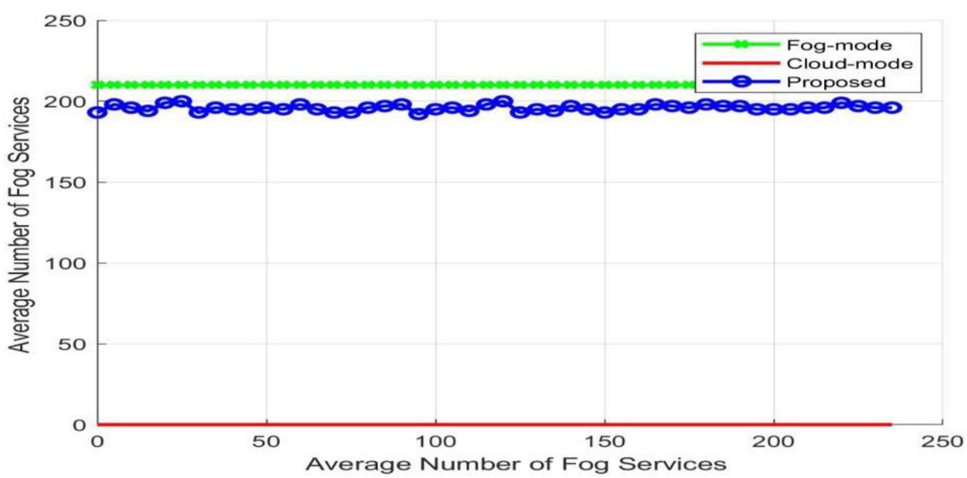

Figure 5-4: Placement of the applications in the fog nodes

In this experiment, the "proposed" approaches were compared to other approaches in fog computing that are presented and analyzed based on the performance criteria of execution time, the number of allocated containers, and the total cost. The observations show that the proposed approach improves the service delay parameter by up to 40%, the total cost by up to 23% and the allocated containers by up to 12%.

5.2. Experiment 2:

In this experiment, the solution proposed in this paper is compared to two methods, "FG" and "OPT" to confirm its effectiveness. For this purpose, Bursty workload for modeling, the requested arrival in 1 day period, and 5-minute intervals were considered. The simulation time is 4 hours and includes 48-time intervals. Besides, the topology settings include three cloud servers, seven cloud nodes, and 30 services.

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

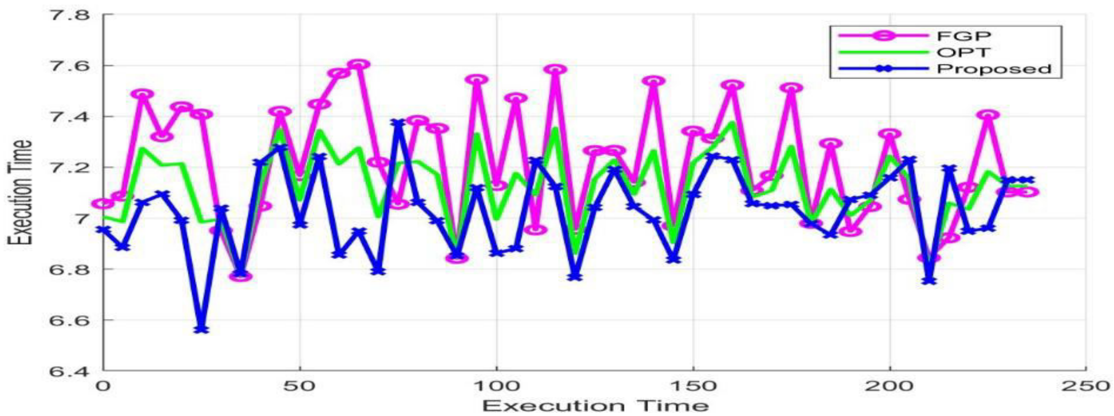

Figure 5-5: Average program execution time

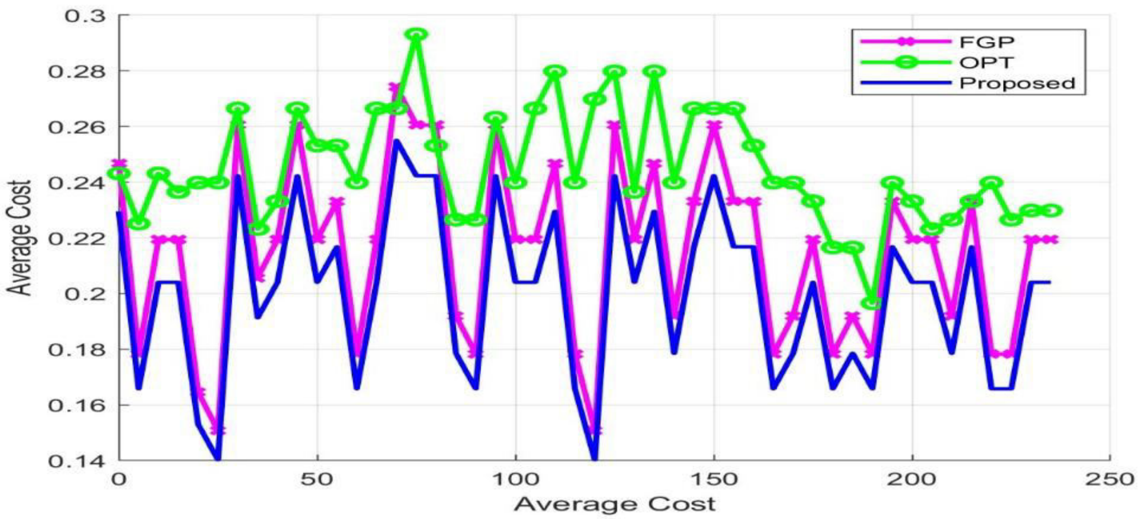

Figure 5-6: Average cost

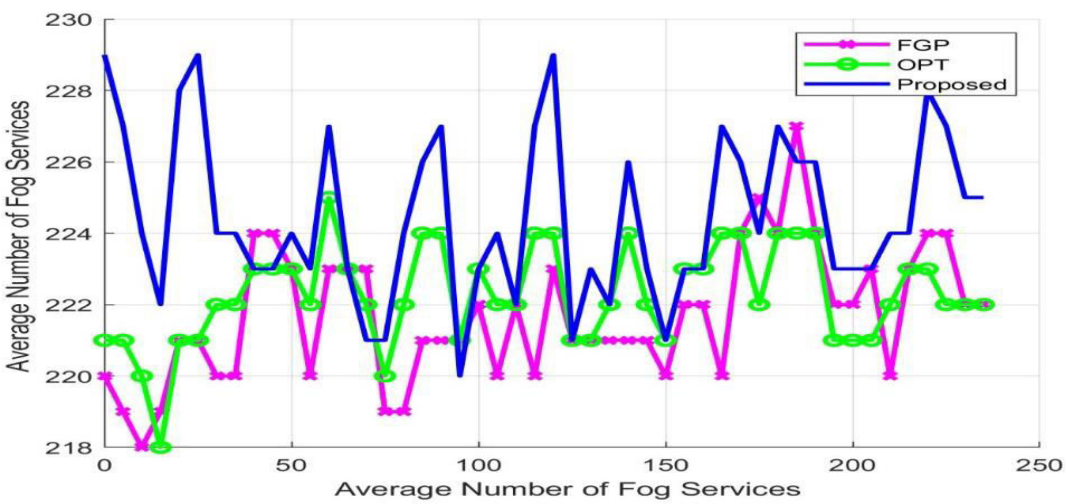

Figure 5-7: Number of Fog services

In this section, the particle swarm algorithm was defined, and the proposed algorithm's development was described.. Diagrams of this algorithm were plotted in this paper and, the use of the system model by this proposed algorithm

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

was described. The computation and formula of the latency time parameters, the cost been specified in this section. In the next section, the findings and results of the simulation of this proposed algorithm using two system models in this research were shown.

6. Conclusion

In this research, the issue of management and placement of services requested by the IOT applications on the fog computing cells was investigated. Fog and edge computing are emerging as an attractive solution to IOT data processing problem. Instead of outsourcing all operations to the cloud, they use the devices on the edge of network with higher processing power instead of end devices, and reduce the network congestion and latency.

In this research, an autonomous management framework for placement fog services was presented. For this purpose, PSO meta-heuristic algorithm was used to locate fog computing cells' fog services. We developed formulas for key performance criteria such as service latency and total system cost and showed how the proposed model determines the appropriate number of fog nodes required for the QOS parameter across all IOT workloads. The simulation results showed that the proposed approach reduced service latency and total cost.

6.1. Recommendations

Due to the results obtained from the present article and for further evaluation of results and completion and development of this research, the following recommendations are presented:

- ✓ Problem formulation and current problem formulation does not include all features and variables in the fog environment due to complexity. The main reason for this is that the standard guidelines and protocols for setting up services on a potential edge node and for communicating between cloud and edge nodes are not fully known and developed, and fog computing is generally in its infancy. Therefore, expanding the problem is one of the future tasks.
- ✓ Due to the various complexities and tasks that each fog cell can have, fog computing can be very non-analyzable and time-varying, making it difficult to analyze it, in general, using simple innovative and meta-heuristic algorithms. Using data mining methods, statistical analysis; finally, new learning algorithms can be very helpful by considering the various elements of cloud computing.
- ✓ Although the PSO algorithm is an appropriate algorithm in optimization problems, but sometimes it is observed that this algorithm is also caught in local optimizations. It is suggested that this problem be considered in future research, and the proposed method of the dissertation be examined with other improved meta-heuristic methods.
- ✓ In recent years, new optimization and capability algorithms such as the penguin algorithm and the Harris hawk algorithm were introduced. It is suggested that in future work, the proposed method of the dissertation be examined with other new algorithms.
- ✓ Development of the proposed solutions in the real world.

These are some items which are less mentioned in this research and need to be discussed and explored in the future as challenging topics.

References

1. Hu, P., et al., Survey on fog computing: architecture, key technologies, applications and open issues. Journal of Network and Computer Applications, 2017. 98: p. 27-42.
2. Asghari, P., A.M. Rahmani, and H.H.S. Javadi, Internet of Things applications: A systematic review. Computer Networks, 2019. 148: p. 241-261.
3. Hashemi, S. and M. Zarei, Internet of Things backdoors: Resource management issues, security challenges, and detection methods. Transactions on Emerging Telecommunications Technologies, 2021. 32(2): p. e4142.
4. Bonomi, F. and R. Milito, Fog Computing and its Role in the Internet of Things. Proceedings of the MCC workshop on Mobile Cloud Computing, 2012.
5. Masdari, M. and A. Khoshnevis, A survey and classification of the workload forecasting methods in cloud computing. Cluster Computing, 2019.
6. Shakarami, A., A. Shahidinejad, and M. Ghobaei-Arani, A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective. Software: Practice and Experience, 2020. 50(9): p. 1719-1759.
7. Dastjerdi, A., et al., Fog Computing: Principals, Architectures, and Applications. 2016. p. 61-75.
8. Ghobaei-Arani, M., A. Souri, and A.A. Rahmanian, Resource Management Approaches in Fog Computing: a Comprehensive Review. Journal of Grid Computing, 2019.
9. Rana, B., Y. Singh, and P.K. Singh, A systematic survey on internet of things: Energy efficiency and interoperability perspective. Transactions on Emerging Telecommunications Technologies, 2020. n/a(n/a).

ISSN- 2394-5125 VOL 8, ISSUE 02, 2021

10.Ghobaei-Arani, M., S. Jabbehdari, and M.A. Pourmina, An autonomic approach for resource provisioning of cloud services. Cluster Computing, 2016. 19(3): p. 1017-1036.
11. Duc, T.L., et al., Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey. ACM Comput. Surv., 2019. 52(5): p. 1-39.
12. Skarlat, O., et al., Optimized IoT service placement in the fog. Service Oriented Computing and Applications, 2017. 11(4): p. 427-443.
13. Mehmandar, M., S. Jabbehdari, and H. Haj Seyyed Javadi, A dynamic fog service provisioning approach for IoT applications. International Journal of Communication Systems, 2020. 33.
14. Mahmud, M.A., A. Abdelgawad, and K. Yelamarthi. Energy efficient routing for Internet of Things (IoT) applications. in 2017 IEEE International Conference on Electro Information Technology (EIT). 2017.
15. Velasquez, K., et al. Service Orchestration in Fog Environments. in 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud). 2017.
16. Mahmud, R., et al., Quality of Experience (QoE)-aware placement of applications in Fog computing environments. Journal of Parallel and Distributed Computing, 2019. 132: p. 190-203.
17. Naranjo, P.G.V., et al., FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments. Journal of Parallel and Distributed Computing, 2019. 132: p. 274-283.
18. Yao, H., et al., Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing. Concurrency and Computation: Practice and Experience, 2017. 29(16): p. e3975.
19. Etemadi, M., M. Ghobaei-Arani, and A. Shahidinejad, A learning-based resource provisioning approach in the fog computing environment. Journal of Experimental & Theoretical Artificial Intelligence, 2020: p. 1-24.
20. Yousefpour, A., et al., FOGPLAN: A Lightweight QoS-Aware Dynamic Fog Service Provisioning Framework. IEEE Internet of Things Journal, 2019. 6(3): p. 5080-5096.
21. Brogi, A., S. Forti, and A. Ibrahim, Deploying Fog Applications: How Much Does It Cost, By the Way? 2018.
22. Ullah, R., M. Atif Ur Rehman, and B.-S. Kim, Design and Implementation of an Open Source Framework and Prototype For Named Data Networking-Based Edge Cloud Computing System. IEEE Access, 2019. PP: p. 1-1.
23. Hanes, D., et al., IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. 2017: Cisco Press.
24. Arkian, H.R., A. Diyanat, and A. Pourkhalili, MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications. Journal of Network and Computer Applications, 2017. 82: p. 152-165.
25. Odun-Ayo, I., et al. Cloud Computing Architecture: A Critical Analysis. in 2018 18th International Conference on Computational Science and Applications (ICCSA). 2018.
26. Sunyaev, A., Fog and Edge Computing, in Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies, A. Sunyaev, Editor. 2020, Springer International Publishing: Cham. p. 237-264.
27. Wang, N., et al., ENORM: A Framework For Edge NOde Resource Management. IEEE Transactions on Services Computing, 2017: p. 1-1.