Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 12, No. 1, February 2021), Pages: 27-50
www.jacr.iausari.ac.ir

An Optimized Method for Radio and Computational Resources Allocation to IoT Users in Fog Computing

Mahmood Lakzaei✉[1], Vahid Sattari Naeini[2], Amir Sabbagh Mollahoseni[1]
1) Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran
2) Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran
mahmood.lakzaei@iauk.ac.ir; vsnaeini@uk.ac.ir; sabbagh@iauk.ac.ir

Received: 2021/02/14; Accepted: 2021/05/02

Abstract

Fog computing is a new approach to evolving the cloud computing platform and extending the Internet of Things to the edge of the network. In this type of computing, service providers can control signals by assigning specific tasks to users.By the substantial increase in the Internet of Things (IoT), the classic centralized cloud computing method has faced several challenges such as high delay, low capacity, and network defects. Fog computing brings cloud waves closer to IoT devices to face with these challenges. The fog provides local processing and storage of IoT data on IoT devices, instead of sending them to the cloud, and provides faster response and better quality in comparison to the cloud The present study was conducted to investigate the allocation of common radio and computing resources to optimize system performance and user satisfaction.in which The effects of cache parameters, CPU clock frequency performance, bandwidth, CPU cycle, and IPC (instructions per cycle / hour) on IoT-Fog users are evaluated to provide a model for allocating radio and computing resources in IoT computing Distribution of computational and radio resource allocation solutions uses a compatible game framework called (SPA). The results of the proposed optimized framework can bring the system performance closer to the desired level from the users' point of view.

Keywords: *Fog nodes, clock frequency, bandwidth, CPU cycle.*

1. Introduction

The Internet of Things is a concept in which all objects are able to interact with each other through communication platforms [1]. In the 21st century, with enormous computational power, high-speed internet, Internet of Things and Blockchain tech-nology, business can be done. It shows that digital contents are widespread on a wide range of con-nected devices. Digital objects such as software, databases, images, audio,videos, and webpages get created and widely distributed over the internet in no time[2]. Due to digitization, a huge volume of data is being generated across several sectors such as healthcare, production, sales, IoT devices, Web, organizations.a Hence, they can be used to that can be used[3].

By the substantial increase in the Internet of Things (IoT), the classic centralized cloud computing method has faced several challenges such as high delay, low capacity, and network defects [4]. Cloud computing has the ability to collect data from user data in one place and then process it to achieve a comprehensive system analysis [5]. Fog computing brings cloud waves closer to IoT devices.Therefore, fog computations are the best choices for activating IoT in order to provide efficient and secure services to several users. Various factors such as service quality, price, justice, profit rate, and load balance play an important role in resource allocation The resource allocation process is very complex since both consumers and cloud providers aim to make the most profit. Any device with processing and computing power in general computation, storage, and network communications could be considered a fog node.In fog computing, data can be processed in smart devices instead of being sent to the cloud, which allows the real-time response to input data and is rendered with bandwidth[4].

Nodes have limitations including memory, reception, communication, as well as the ability to calculate and transfer large amounts of additional data [6]. Allocating Fog Nodes (FNs) with limited computational resources to all DSSs is regarded as an essential issue in achieving optimal and stable performance. The present study proposes a common optimization framework for all DSSs, DSOs, and FNs to achieve optimal resource allocation schemes in a distribution method, in which a Stackelberg game is used to analyze the resource allocation problem for DSSs problem[7].

In this paper, a matching game framework, called Student Project Allocation (SPA), is used to distribute the solutions of allocating computational and radio resources, instead of the usually centralized optimization. In addition, the proposed User-Oriented Cooperation (UOC) strategy eliminates the external effects of instability, i.e. the interdependence between identical players. Finally, the system efficiency is measured after using UOC strategy and comparing with that of the previous study [8]. also In this research, influential parameters in both areas (radio & Computational) are identified to fill in the existing gap and present an applied model. Furthermore, the suggested approach in [8] is improved by considering the joint optimization of energy consumption and performance of delay (EDM).The effects of different parameters such as delay, productivity,SP Profit, cost-effectiveness, bandwidth, user access time, cache memory performance, and CPU clock frequency on the allocation of joint computational and radio resources in IoT-fog computations are examined and Some essential factors including mandatory advantage, link quality, and service delay are considered.

One advantage of this paper than other similar studies is that the delay is much less (1.4 seconds) than other methods and the main advantage of this article is that all factors affecting the allocation of computing and radio resources have been considered. This issue has not been addressed in any of the previous articles and only one of the factors influencing computing has been studied.

In this article, the SPA- (S, P) and UOC algorithms were used to identify important and influential parameters and provide a functional and optimal model for allocating resources in joint radio and computational of IoT-fog computing and investigating the impact of some factors. each factor such as “Service delay”, “Service quality” based on control signals and “Bandwidth”, allocated and optimized “CPU cycle” performance relying on effective and efficient factors in the face of possible requests and fast computing.

2. Related works

For such applications, for instance Zeng et al[9] introduced a schedule-based load allocation policy to equalize the computational load in client devices and on fog nodes. Based on the results of this study, the allocation of workload in fog computations should be conducted in such a way that the utilization of resources is optimized and unemployment time intervals are minimized.

Taneja and Davya[10] presented an analytical model for running on a cloud,or fog computing by developing a knowledgeable source of the foundation on data analysis in cloud computing architecture. The results showed that computational applications with a focused approach on cloud computing require more reduction of reaction time in delayed-sensitive applications, which can minimize network costs and increase efficiency using fog calculations .Lewis et al[11] presented the resources supply methods for tactical cloudlets to provide an infrastructure which supports assigning computations and storing the data at the tactical edge. They developed various policies to provide virtual machines on cloudlets.

Zhang et al[12] evaluated a special fog computing network composed of a set of decision support system operators, every single of which controlled a set of FNs and the necessary data services were provided to a set of subscribers. A common optimization framework was proposed for all FNs, DSS decision support, and modulation system to develop a plan to allocate optimal resources in a distributed model.

Yousefpour et al[13] focused on the improvement of service quality in IoT networks and introduced a general framework for IoT cloud applications. They proposed the policy of minimizing delays for fog-devices to reduce service delays in IoT applications.

Ruilong et al[14] examined the relationship between transmission delay and power consumption in a cloud computing system and offered a method for optimal load allocation between fogs and clouds with minimal power consumption and limited-service delay. The results showed that using moderate computational resources to maintain bandwidth, communication, and reducing fog calculations transmission could significantly improve the cloud computing performance.

Dinh et al[15] proposed a framework for optimizing the offloading from one mobile device to several other devices. They minimized performance delays and energy consumption and optimized decision-making in task allocation and performance in the Central Processing Unit (CPU).

Suzhi Bi et al[16]considered a multi-user caching-enabled MEC system, where users with their task requests proactively cached and executed at the edge server can directly download the desired results. In FDMA setup, jointly optimized the cache placement and the BW allocation to minimize the weighted-sum energy of the edge server and the users subjected to the computation, communication and caching capacities as well as the service latency constraints.

Xi li etal[17]focused on the optimization of computation and communication resource allocation in fog computing-based wireless IoT networks with NOMA.and considered a general scenario with massive IoT devices, and modeled the cost and energy consumption for both local computing and offloading computing tasks to FN.they found that the system energy consumption and the average delay could be impacted by the different computing modes. the proposed make an optimal decision for choosing the proper computing mode could achieve a good performance.

Ghonoodi [18] presented a new scheduling algorithm called Energy Aware Scheduling Algorithm based on DVFS technique and task duplication strategy. simulations show that this algorithm has a good improvement on energy efficiency for parallel applications.

Yingteng Ma et al[19] considered the joint resource allocation of computing and communication in F-RAN. order to satisfy the QoS of users of different types of tasks, formu-lated the MINLP problem with the limited processing power and fronthaul of each F-AP. genetic convex optimiza-tion algorithm considering user access, computing of oad, computing resource allocation, and spectrum resource allo-cation is proposed to obtain a feasible suboptimal solution.The simulation results showed that the genetic convex optimization algorithm has a better payoff than than discrete.

Xiaoge Huang etal[20] studied resource allocation problem in fog computing networks by the candidate FN mechanism. In the scenario, FNs could decide whether to join in the task offloading scheme and the associated computation capability allocated to IoT devices accord-ing to the historical status and the current status. The FN which reports nonzero computation capability was considered as the candidate FN and included in the can-didate set. A candidate FN based resource allocation algorithm (CF-EE) was proposed to maximize network EE under various constraints, which was converted into the Lyapunov optimization for each time.

Praveen Kumar et al [24] designed a hybrid Whale Optimization Algorithm-Moth Flame Optimization (MFO), to select optimal CH, which in turn optimizes the aforementioned factors. The performance of the proposed work is then evaluated with existing algorithms with respect to the energy-specific factors. The results obtained prove that the proposed method outperforms existing approaches

Vinay Kumar Billa et al[25], proposed a system which mines frequent itemsets in an optimized way in terms of memory and time by using cloud computing as an important factor to make the process parallel and the application is provided as a service. A complete framework which uses a proven efficient algorithm called FIN algorithm. FIN algorithm works on Nodesets and POC (pre-order coding) tree. In order to evaluate the performance of the system authors conduct the experiments to compare the efficiency of the same algorithm applied in a standalone manner and in cloud computing environment on a real time data set which is traffic accidents data set. The results show that the memory consumption and execution time taken for the process in the proposed system is much lesser than those of standalone system.

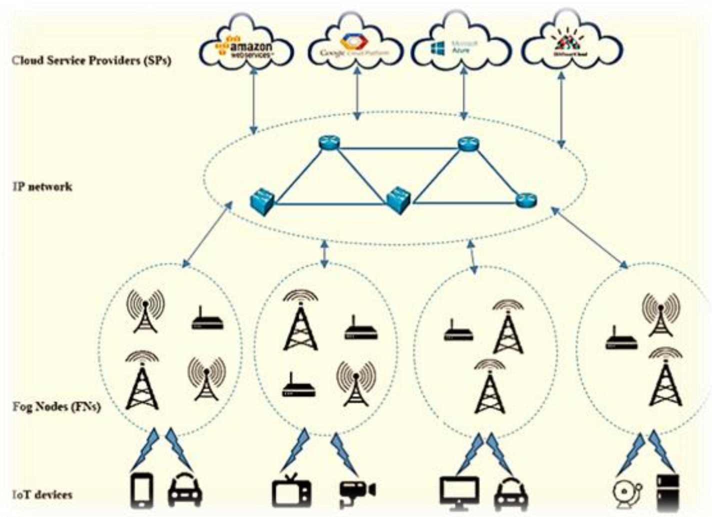
G. Thippa Reddy et al[26], proposed a novel security management framework by employing data mining to detect, contain and prevent attacks on cloud computing systems.

Rajesh Kaluri et al[27], focuses on the predictions pertinent to the sustainability of battery life in IoT frameworks in the marine environment. Firstly, the missing values in the data are replaced with the attribute mean. Later, one-hot encoding technique is applied for achieving data homogeneity followed by the standard scalar technique to normalize the data. Then, rough set theory is used for feature extraction, and the resultant data is fed into a Deep Neural Network (DNN) model for the optimized prediction results. The proposed model is then compared with the state of the art machine learning models and the results justify its superiority on the basis of performance metrics such as Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, and Test Variance Score.

Gu et al.[8] examined the allocation of joint computational and radio resources by considering essential factors such as power transmission, transmission quality, and service delay. They utilized the framework of matching theory, particularly the SPA game, to model the problem and applied the SPA- (S, P) algorithm to discover a stable result for the formulated problem.The present study aims to complete, develop, and evaluate the proposed algorithm by Gu and et al.[8] to provide a model for allocating both radio and computational resources in processing through increasing the services of IoT users and relying on effective parameters in increasing data processing speed. In addition, various aspects of increasing the efficiency and quality of services in communication systems are considered.

3. System Model

As shown in figure 1, the networks including loT devices such as smartphones are defined $u = \{u_1, u_2, \dots u_m\}$ as IoT users. A certain type of computing or storage tasks, defined as $sp = \{sp_1, sp_2, \dots sp_n\}$, may be offloaded by users for cloud Service Providers (SPs). These SPs can satisfy the specific computing needs of different users' service delays with regard to data size.Computations are delivered to the cloud for those users which are not sensitive to delays while SPs allocate one of the adjacent FNs to offload the computational task for users with precise delay requirements. FNs closer to users usually lead to lower transfer delays. However, the geographical location affects the overall service delay [8]. The service delays consist of three time periods including CPU processing time, reception time, and transfer time. Transfer and reception times are respectively used to send data to FNs for processing, and the time applied to receive the processed results. Such communication delays are releted to channel conditions and affect the size of the computational data. The CPU processing time is determined by the CPU rate of every single FN. Therefore, for each SP, sp_j jointly allocates its radio resources $W^j = \{w_1^j, w_2^j, \dots w_k^j\}$ (channel bandwidth) and computing resources $C^j = \{c_1^j, c_2^j, \dots c_L^j\}$ (CPU cycle rate) during the selection of the appropriate FN f $FN^j = \{fn_1^j, fn_2^j, \dots fn_L^j\}$ or each user.



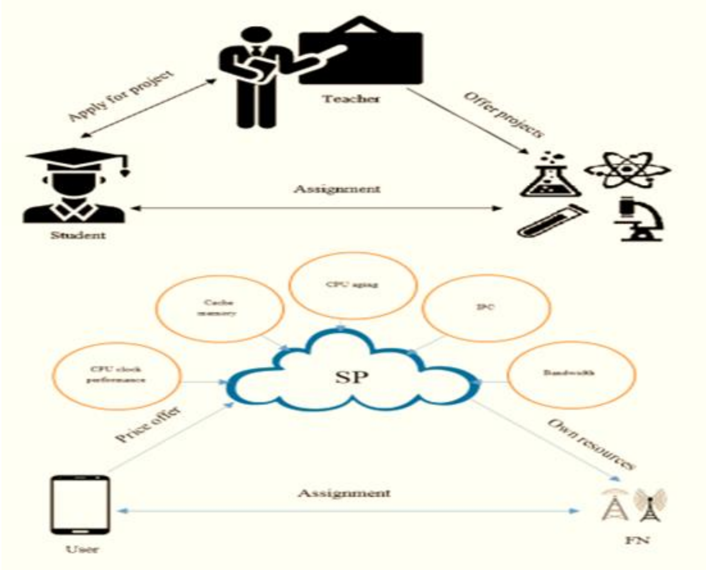


Figure 1. A system model

From the users' perspective, anyone with sensitive content which is delayed in processing offers prices to SPs to compete for better radio and computing resources. The users requiring less delay have an intuitive tendency towards higher prices. Furthermore, the users consider the data size because a larger data size usually takes a longer time to process and requires longer CPU processing time. In the present study, the CPU cycle for data processing is as dependent as the data but not exactly the same. Therefore assume that each u_i user has D_i (bits) data, a DC_i processor cycle is needed for the related processing task, and there is a linear relationship between DC_i and D_i to simplify the subject [17].

The allocation of joint radio and computing resources could be considered as a mapping among the user of u and (radio, computational) $\{(\omega_k^j, c_1^j)|\forall\omega_k^j\epsilon w^j, c_k^j\epsilon c^j\}$ resources belonging to $sp_j, sp_j \in SP$ of each SP. For simplicity, $rp_{l,k}^j$ is used to specify (ω_k^j, c_1^j) . It is shown that the relationship between mapping and $p_{k,l}^{i,j}$ binary value is $p_{k,l}^{i,j}=1$ if the u_i is offloaded to the fn_l^j FN using the ω_k^j channel belonging to the sp_j, and $p_{k,l}^{i,j}=0$ otherwise. As shown in the following section, the benefits of both user and SP are considered for optimizing the joint resource allocation.

3.1. Services delay measure of user satisfaction

User satisfaction or user experience is considered as one of the essential metrics that all SPs are concerned about. As discussed earlier, a set of users with sensitive delay conditions is discussed. Therefore, the service delay is applied as a measure of user satisfaction. However, before discussing the delay, it must be ensured that the transmission quality between FNs and users is capable of meeting this requirement. In other words, the Signal to Interference Noise Ratio (SINR) must be higher compared with the Γ_{min} threshold in order to provide accurate/complete data. The SINR received from u_i is defined using w_k^i, as Eq (1):

$$\Gamma_{k,1}^{i,j} = \frac{P_i g_{k,1}^{i,j}}{\sum_{u_{i'} \in u, i' \neq i} \rho_{k,1}^{i',j} P_{i'} h_{k,1}^{i',j} + \sigma_N^2} \tag{1}$$

where P_i and $g_{k,l}^{i,j}$ are the power of transmission and channel increase between the u_i user and fn_l^j fog node using the w_k^j channel. The $h_{k,l}^{i',j}$ shows the interference benefits of the channel from any other $u_{i'}$ mobile user in fn_l^j because of re-using the channel. It is assumed that orthogonal radio sources applied among SPs and radio resources in the SP can be coordinated to prevent the interference. σ_N^2 indicates channel noise. To ensure a successful transfer, $\Gamma_{k,1}^{i,j} \geq \Gamma_{min}$ should be considered.

If SINR requirements are met, the transfer rate from u_i using w_k^j should be based on Eq. (2) as follows:

$$r_{k,1}^{i,j} = w_k^j \log\left(1 + \Gamma_{k,1}^{i,j}\right), \tag{2}$$

As mentioned earlier, service delays include three time periods, namely, transfer time (t_{trans}), CPU processing time (t_{proc}), and reception time (t_{recv}). In general, the results obtained from FN after processing is insignificant compared to the original unprocessed data. In addition, it is not possible to predict the accurate size of the returned data without understanding the results after processing. Therefore, the receiving time must be short enough, and it is assumed that the $\delta t, \delta t \in [0,1]$ variable is random for each user to show trecv. The channel should be re-used, and the CPU should be shared between multiple users to define t_{trans} and t_{proc}. The share of every single channel among more than one user in q_R capacity and the accommodation of more than one user by each FN to share their q_C CPU capacity are permitted by the authors. Moreover, q_{SP} is the maximum number of users that an SP can serve. Therefore, the interference of shared channel users can affect the transfer rate for each user, as shown in Eq. (1). Furthermore, the shared CPU users affect the CPU processing rate. For simplicity, it is assumed that every single joint CPU user will receive an equal share of the total CPU rate as $c_{k,1}^{i,j} = \frac{1}{\sum_{u_i \in u} \rho_{k,1}^{i,j}} c_1^j$. Now, the u_i service delay can be defined by using the $\left(w_k^j, c_l^j\right)$ source pair as Eq. (3).

$$t_{k,l}^{i,j} = t_{trans} + t_{proc} + t_{recv} = \frac{D_i}{r_{k,l}^{i,j}} + \frac{DC_i}{c_{k,l}^{i,j}} + \delta t. \tag{3}$$

3.2. SP revenue for system performance measurement

Mandatory revenue is a motivating factor which allows SPs to provide better services to their shared users. Additionally, price offers from users are used for SP profit/revenue as another factor for measuring the system performance. As previously discussed, the price offered by each user depends on not only T_i delay but also D_i size. It is assumed that there is a linear relationship between data size and price, and the inversion of delay. Therefore, a suggestion from any user can be provided as Eq (4).

$$O_i = f(D_i, T_i), \tag{4}$$

where f (.) must be a monotonic increase function for D_i and the monotonic reduction function for T_i. For simplification, Eq (5) is used to define $f(D_i, T_i)$.

$$O_i = a\frac{D_i}{T_i}, \tag{5}$$

where 'a' denotes a parameter in dollar / Mbps unit, and O_i is the price which u_i should pay per SP if matched.

Every single SP serves more than one user. Thus, it receives more than one offer. The revenue sp_j of SPs is defined as the sum of the mandatory offers collected from all adapted users. In the present paper, the cost of every single SP is associated with electricity consumption for maintenance and transmission, which are assumed to be constant for the sake of simplicity. The impact of fixed service costs is ignored when considering the SP income. Consequently, the total revenue of each SP is defined as Eq (6):

$$Reu_i = \sum_{u_i \in u} \rho_{k,l}^{i,j} O_i. \tag{6}$$

4. problem formulation

In the previous section, two essential performance metrics for allocating appropriate resources in fog computing were discussed. The proposed system in this paper aims to provide the combination of both designed criteria, called cost-performance (CP), which is defined as the ratio between the average data rate per user and its cost per unit Mbps / sec/dollar. The rate of history is considered instead of the net delay since the amount of real delay is significantly associated with the size of the user data which should be transferred and processed.

Therefore, if compared horizontally with other users, the actual data rate is a more appropriate criterion in comparison to the delay rate. This issue can be considered for the cost factor and the use of paid payment/user suggestions for the related fog service. Consequently, to combine the two factors into one metric, the cost-performance function is defined for every user, physically indicating the quality of the service for which the user pays. The CP_{sys} CP system is the average CP of all users $CP(i)$, as Eq (7).

$$CP_{sys} = \frac{\sum_{u_i \in u} CP(i)}{M}, u_i \in u, \tag{7}$$

where $CP(i)$ is defined as Eq (8) and refers to the CP value for the u_i user.

$$CP(i) = \rho_{k,l}^{i,j} \frac{\frac{D_i}{t_{k,l}^{i,j}}}{O_i}. \tag{8}$$

In the next step, the optimization problem is adjusted based on the (9-15) equations.

$$\max_{\rho_{k,l}^{i,j}} : \frac{\sum_{u_i \in u} CP(i)}{M} \tag{9}$$

$$\text{s.t.}: \rho_{k,l}^{i,j} t_{k,l}^{i,j} \le T_i, \quad \forall u_i \in u, rp_{l,k}^{j} \in RP^{j}, sp_j \in SP, \tag{10}$$

$$\rho_{k,l}^{i,j} \Gamma_{k,l}^{i,j} \ge \Gamma_{\min}, \quad \forall u_i \in u, rp_{l,k}^{j} \in RP^{j}, sp_j \in SP, \tag{11}$$

$$\sum_{u_i \in u, fn_l^j \in FN^j} \rho_{k,l}^{i,j} \le q_R, \forall w_k^j \in W^j, sp_j \in SP, \tag{12}$$

$$\sum_{u_i \in u, w_l^j \in W^j} \rho_{k,l}^{i,j} \le q_C, \forall fn_l^j \in FN^j, sp_j \in SP, \tag{13}$$

$$\sum_{u_i \in u, rp_l^j \in RP^j} \rho_{k,l}^{i,j} \le q_{SP}, \forall sp_k^j \in SP, \tag{14}$$

$$\rho_{k,l}^{i,j} \in \{0,1\}, \tag{15}$$

Eq (9) is the purpose of the system and shows the performance of the total cost for users. The minimum SINR requirement for every single user is defined by Eq. (11). The Eqs. (12), (13), and (14) meet the capacity limit for every single channel, FN, and SP, respectively.

This optimization is an MINLP issue, which is NP-hard in general Hence, a suboptimal practical solution should be suggested. Therefore, a distributed theory-based method called Student Project Allocation game is introduced and discussed in the following section.

5. Student-project matching game

Allocating radio and computing resources allows us to consider the pair (radio, computational) as a single entity. The possible combinations of two kinds of resources could be counted to draw the maps of the user set to the source pairs. It seems the counting and mapping process should be monitored by SPs, which are responsible for communicating the control signal to both users and sources.An appropriate implementation model providing exactly such structures is Student Project Allocation (SPA) [23], in which different projects are allocated to different students with the support of different teachers. In this section, the modelling of the proposed problem is introduced using the SPA model, and then the SPA- (S, P) algorithm is presented to implement a sustainable matching solution [8]. However, a cross-channel collaboration strategy is recommended to eliminate the external effects and ensure system stability to deal with the external consequences during implementation.

Definition 1. Stability: The matching M is called stable if there is no Blocking Pair (BP). A $(u_i, rp_{l,k}^{j})$ pair is defined as BP if all of the following conditions are met:

(1) u_i, considers $rp^j_{l,k}$ acceptable;

(2) Whether u_i does not correspond to M and whether u_i prefers $rp^j_{l,k}$ to $M(u_i)$.

(3) Whether

(3.1) $rp^j_{l,k}$ has been confirmed that each of the following three conditions must be met:

A) $M(u_i) \in RP^j$ and sp_j, prefer $(u_i, rp^j_{l,k})$ to $(u_i, M(u_i))$, or

B) $M(u_i) \notin RP^j$ and sp_j approved, or

C) $M(u_i) \notin RP^j$ and sp_j is complete and sp_j prefers $(u_i, rp^j_{l,k})$ to the worst current pair (u_{wst}, rp^j_{wst})

(3.2) $rp^j_{l,k}$ is complete and sp_j prefers $(u_i, rp^j_{l,k})$ to the worst current pair (u_{wst}, rp^j_{wst}), and each of the following two conditions must be met:

A) $M(u_i) \notin RP^j$

B) $M(u_i) \in RP^j$ and sp_j prefers $(u_i, rp^j_{l,k})$ to $(u_i, M(u_i))$.

In definition 1, $M(x)$ represents the partner/match player x following M. More precisely, $M(u_i) = r, p^j_{l,k}, (w^j_k, c^j_l) \in RP^j$.

In order to find consistent compliances, users' preferred lists and SPs must first be created as PL^{user} and PL^{SP}, respectively. In this method, the limitations of equations (10) and (11) must also be considered in terms of user delay. Furthermore, the SINR should be met. In other words, each user must first consider these two limitations when setting up preferential lists for users and consider the pairs of resources. These pairs of resources are called acceptable sets. After finding acceptable collections for all users, the authors rank the pairs of resources as descending/ascending for each user according to their preferences [8].Visually, users prefer resources that can deliver computing loads with the least delay. However, the coexistence of multiple users affects the radio and CPU performance since each pair of sources is allowed to accommodate more than one user. For simplification, these users equally share the same coexistence, frequency band, and CPU rate. Therefore, the number of available and shared resources determines the share of each user. This number is not specified for the user and SP before the matching is finalized. However, each SP and every radio and CPU resource have $q^{SP} q^R$ and q^C symbols, which limit the maximum number of users. In order to calculate the potential service delay, each user shares $\frac{1}{Q}$ of the radio and CPU resources, depending on the exact quota. The actual performance may deviate from this assessment, which has an external effect during implementation. Therefore, the priority of each u_i user on $rp^j_{l,k}$ is based on the $t'^{i,j}_{k,l}$ potential service delay and is provided as Eq. (16):

$$PL_i^{user}(j,k,l) = t'^{i,j}_{k,l} = t'_{trans} + t'_{proc} + t'_{recv}$$
$$= \frac{Di}{\frac{1}{q^R} r'^{i,j}_{k,l}} + \frac{DC_i}{\frac{1}{q^R} c_l^j} + \delta t', \tag{16}$$

where $r'^{i,j}_{k,l}$ is the data rate from u_i to fn_l^j FN when u_i only uses the w_k^j channel and is displayed as $r'^{i,j}_{k,l} = w_k^j \log(1 + \frac{P_i g_{k,l}^{i,j}}{\sigma_N^2})$. $\delta t'$ is another random value inside [0,1] which indicates the possible time period for the return of the result. On the other hand, when picking users to match a pair of resources, SPs not only consider the mandatory benefits associated with the measure, as well as the potential delay in the service. The delay coefficient works similarly for users and SP since users expect the service to be faster. SPs follow the short service time per user to more service users in a long time. Therefore, SP preferences for users are based on the price-to-earnings ratio (the same as potential delay estimation for users) and are presented as Eq. (17):

$$PL_{j,k,l}^{SP}(i) = \frac{O_i}{t'^{i,j}_{k,l}}. \tag{17}$$

By setting preferential lists, the SPA- (S, P) algorithm could be used for finding an efficient match between users and resources, as shown in Algorithm 1. The original idea for the SPA- (S, P) algorithm was developed from the Gale-Shapley classic algorithm [21].

5.1 SPA- (S, P) Algorithm Definition

This algorithm includes consecutive suggestions and acceptance/rejection operations by users and SP.As shown, sustainable matching is guaranteed under the central matching condition, which means that the players' preference does not depend on other players' selection/performance, but on local information about other types of players. Therefore, matching the implementation of the SPA- (S, P) algorithm is not necessarily stable and requires more measures to achieve stability. In the present study, a one-way collaboration is proposed to make the current matching sustainable.

5.2 User-Oriented Cooperation Strategy

The SPA- (S, P) algorithm matching is not necessarily stable due to the internal dependence on user and resource preferences affected by the existing matching.For example, an excellent previous resource pair may be evaluated by many users sharing it, which could not be appropriate for a considerably small number of users sharing it. There may be incentives for users to exchange for other resources, which, in turn, become BP. Some algorithms can be designed to remove these BPs. However, it makes sense to think more from a user's perspective. At this time, the user begins to evaluate the concept of stability only by the user through evaluating the purpose of the proposed system and calculating the average cost of user performance. In other words, it is assumed that only the users are motivated to make changes.

Algorithm 1 *SPA-(S,P) Algorithm*
Input: U; SP; W; FN; $P\mathcal{L}^{user}$; $P\mathcal{L}^{SP}$;
Output: Matching M;
Initialization: set M empty, set all users free;
while some user u_i is free and u_i has a non-empty preference list **do**
  for all $u_i \in U$ **do**
    u_i proposes to the first entity $rp_{l,k}^{j}$ in $P\mathcal{L}_i^{user}$, and then remove $rp_{l,k}^{j}$ from $P\mathcal{L}_i^{user}$;
    $M \leftarrow M \cup (u_i; rp_{l;k}^{j})$;
  end for
  for all $rp_{l,k}^{j}$, $rp_{l,k}^{j} \in RP^{j}$; $sp_j \in SP$ **do**
    while $rp_{l,k}^{j}$ is over-subscribed **do**
      Find the worst pair (u_{wst}, rp_{wst}) assigned to $rp_{l,k}^{j}$ in sp_j's list;
      $M \leftarrow M / (u_{wst}, rp_{wst})$;
    end while
  end for
  for all $sp_j \in SP$ **do**
    while sp_j is over-subscribed **do**
      Find the worst pair (u_{wst}, rp_{wst}) in sp_j's list;
      $M \leftarrow M / (u_{wst}, rp_{wst})$;
    end while
  end for
end while
Terminate with a matching M.

Therefore, a new 'stability' symbol, which is different from Definition 1 and relies on a balance between all users, must be defined between the users. The cooperation between users is necessary to convert the existing matching into sustainable matching.The definition of optimal Pareto is as follows:

Definition 2. Optimal Pareto: A match is called Optimal Pareto if there is no other match in which some players are better the others while no player is in a worse situation. Accordingly, the new BP definition for the one-way matching problem is presented in Definition 3.

Definition 3.

A BP in unilateral compliance:

A pair of users is defined as BP if ui and uj are better after exchanging their partners.In order to find such an optimal Pareto compliance, the users re-use SP to evaluate the tools. In Definition 2, Optimal Stability / Pareto is achieved when there is no better player/user without worsening the other player(s).Through such limited exchanges, a stable system without exchange can be finally obtained. The details are presented in Algorithm 2.

Algorithm 2 *User-Oriented Cooperation (UOC) Strategy*

Input: Existing matching M_0;

Output: Pareto optimal matching M_s.

$M_t = M_0$;

while M_t is ”unstable" (user, user) pairs BP **do**

    for all $(u_{i1}, u_{i2}) \in BP$ **do**

        if $\exists u \in M_t(rp_{i1}) \bigcup M_t(rp_{i2}), \Delta \bigcup(u) < 0$ **then**

            (u_{i1}, u_{i2}) are not allowed to switch partners;

        else

            $(u_{i1}; u_{i2})$ are allowed to switch partners;

        end if

    end for

    Find the optimal BP $(u_{i1}^*, u_{i2}^*) \in BP$;

    u_{i1}^* and u_{i2}^* switch partners;

    $M_{t+1} \leftarrow M_t / \{(u_{i1}^*, M_t(u_{i2}^*)), (u_{i2}^*, M_t(u_{i2}^*))\}$;

    $M_{t+1} \leftarrow M_t \bigcup \{(u_{i1}^*, M_t(u_{i2}^*)), (u_{i2}^*, M_t(u_{i1}^*))\}$;

    Update PL^{user} based on M_t;

end while

$M_s = M_t$.

In Algorithm 2, $rp_{il} = M_t(u_{i1}), rp_{i2} = M_t(u_{i2})$. U (x) is defined as a function of the x user tool, which is equal to its service delay. Therefore, $\Delta U(x) = U(x)' - U(x)$, where $U(x)'$ is the tool after the exchange of partners. In other words, $\Delta U(x)$ represents a change in user X performance. $\Delta U(x) > 0$ means an improvement or $\Delta U(x) < 0$ means a decrease. A pair of users are allowed to switch to their partners if and only if $\Delta U(x) \geq 0$ for each x user who is affected by this switch (e.g., $\forall x \in M_t(rp_{i1}) \bigcup M_t(rp_{i2})$). Then, the authors search for a BP which provides the highest performance to find the optimal BP among all the BPs allowed to partner switches. Performance means the same long delay for all users. The optimal BP is defined as Eq. (18):

$$(u_{i1}^*, u_{i2}^*) = \arg\max_{(u_{i1}, u_{i2})} \sum_{u \in \{u_{i1} \cup u_{i2} \cup M_t(rp_{i1}) \cup M_t(rp_{i2})\}} \Delta U(u), \tag{18}$$

The (u_{i1}, u_{i2}) user pair must be allowed to exchange partners. The steps of the UOC strategy could be summarized as follows:

First, search for all ‘unstable’ user exchanges (motivated by exchanges) on the current match. Second, check whether switching is allowed between these pairs (useful for related users). Third, find the right pair, which offers the most progress to change your partners and bring the current match up. Then, continue searching for such BPs until reaching a free trade network. The convergence of the UOC process is ensured by the irreversibility of each switch. Eventually, the UOC ends up with optimal Pareto compliance, which improves the system power at the same time[8].

5.3 proposed method

Data from the CloudHarmony[1] database is used to implement the proposed research model. In this database, there are six public clouds and more than 100 samples of virtual machines, in which the data related to the samples of the machines with computational optimization are used.

In this paper, the authors evaluated both the SPA- (S, P) algorithm and the UOC strategy for w.r.t. user service delay, SP profits, and system cost performance. In addition, the UOC convergence was analysed. Additionally, the authors considered a network with N = 2 cloud service providers, each equipped with L = 5 fog nodes which are randomly distributed in the network with a radius of R = 1 km. Assuming many IoT (M) users, M∈ [45,210] is also randomly distributed on the network. Each SP has K = 5 channel bands for user sharing, and bandwidth is set to w = 5 MHz. The SINR Γ_{min} requirement for users is a uniform random distribution in $[20{,}30]\ dB$. For each channel and each FN, the authors set an equal capacity, which is $q_R = q_C = 10$, and the SP capacity is set to $q_{SP} = 80$ for each. The delay rate, data size, and the corresponding CPU cycle are determined by specific types of IoT devices.

Service delays include delays in transmission and CPU processing delays, and the need for T_d total delays for each user is evenly distributed in [8]. D users’ data size is set to evenly distributed, $[2{,}8]\ dB$, and the corresponding CPU cycle is determined as $DC_i = D_i * 10^4$ 4 cycles.

The CPU processing rate for each FN is determined as a uniform distribution in $[5{,}6] * 10^{10}$ cycles/second. In order to increase the propagation g, the drop in the stable path C as 10^{-2} is determined. The power of the α path decreases by 4. The fading of several paths increases as the unit distribution is averaged, and the shadow increases as the normal log distribution increases by 4 dB.To compare user and SP performance, a random method is used as a victim strategy, which refers to the allocation of random resources between the users and pairs of resources. In addition, the proposed method in [22] is modified by considering the common optimization of energy consumption and performance delay (EDM).

Http://cloudharmony.com [1]

In order to assess user satisfaction, the proposed method in [8] is modified, and the interaction between SP and users is analysed using the Stackelberg game. Further, the authors evaluated system cost performance in three ways.

The cost performance is a benchmark for evaluating the cost of providing services. In this paper, influential parameters in both areas (radio & computing) are identified to fill in the existing gap and present an applied model. In order to allocate computational and radio resources, significant factors including the service quality based on controlling signals, service delay, amount of allocated bandwidth, optimized CPU performance by relying on effective and efficient factors in the face of possible requests,considered as a key factor in rapid computations.Furthermore, the suggested approach in[9] is improved by considering the joint optimization of energy consumption and performance of delay (EDM).The effects of different parameters such as delay, productivity,cost-effectiveness, bandwidth, user access time,cache memory performance,and CPU clock frequency on the allocation of joint computational and radio resources in IoT-fog computations are examined.

5.4 Simulation tool

Simulation in the CloudSIM environment based on iFogSIM, CloudSDN, and CloudEdge packages is performed in the integrated Net Beans IDE software. The network information should be analysed to evaluate the proposed approach, and the benchmark analysis and data set can be obtained from the CloudHarmony website. CloudHarmony is used to monitor the availability of Cloud services (38 in total) from late 2009 to January 2020. Domain monitoring providers are responsible for providing IaaS services such as GoGrid, Rackspace, and Amazon web services as well as PaaS services such as Microsoft Azure and Google App Engine. Different parts of this data set can be viewed, which examines different servers in which a series of main parameters including RAM usage, CPU usage, available and used storage memory, a type of architecture in terms of 32 or 64 bits determine the use of inputs and outputs and their costs.

6. Performance evaluation

CloudSIM software and other software packages, along with their formwork, are integrated environments which are analysed, the results of which are displayed on NetBeans command lines, and the required results are compared with those of [5]. In the present study, SPA- (S, P) and UOC algorithms are used for evaluating the effect of cache parameter on resource allocation to IoT users, CPU cycle parameter on resource allocation to IoT users, and access time parameter on resource allocation to IoT users. The main simulation settings are shown In Table 1.

Table 1: The main simulation settings

Cloud service providers (SPs)	Fog nodes	Radius (Km)	Channel bands	SPs capacity	Bandwidth (MHz)	Data size per request (bytes)	Request per user per Hr
2	5	1	10	80	5	100	12

In this section, the simulation results of the proposed method on the benchmark data collection of CloudHarmony website in Net Beans IDE software using two SPA- (S, P) and UOC algorithms are presented to investigate the effects of CPU clock frequency performance, aging parameter, IPC, cache

performance, and bandwidth on resource allocation to IoT users. Figure 2 shows the results of user response time and access.

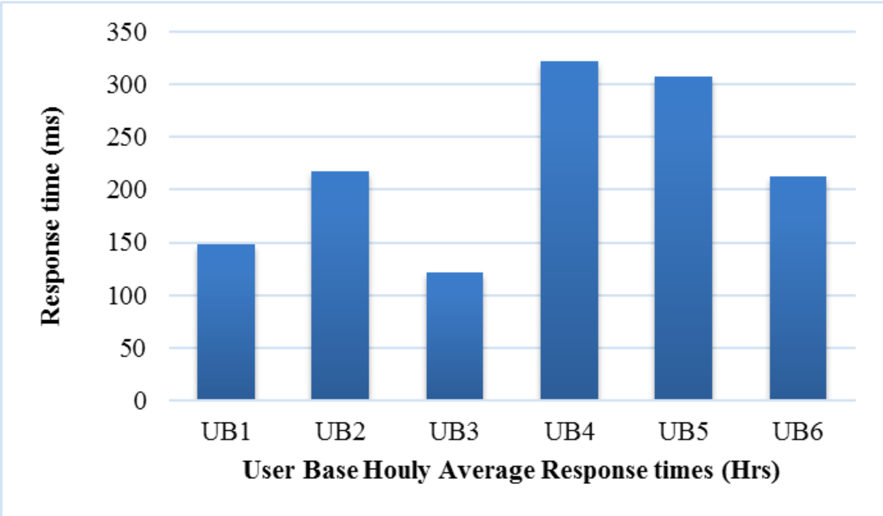


Figure2. Results of response time and access

6.1. Service Delay

In figure 2, the average service delay assessment is compared by using five methods, namely, random method, EDM algorithm, SPA- (S, P) algorithm, SPA- (S, P) with UOC strategy, and the proposed approach. The optimal method is SPA- (S, P) with UOC strategy and all 220 users. Gu et al. [8] compared service delay with four random methods including EDM algorithm, SPA- (S, P) algorithm, SPA- (S, P) with UOC strategy, and random. In the present study, the number of users increases from 45 to 220 to indicate a change in delay. Service delays for all five strategies increase with the number of the users, which could be justified by considering the fact that more users cause lower average resources per person, which leads to more delays. As shown, the average service delay of the proposed approach (SPA-(S, P) and UOC with 220 users has a lower rate compared to the previous methods.

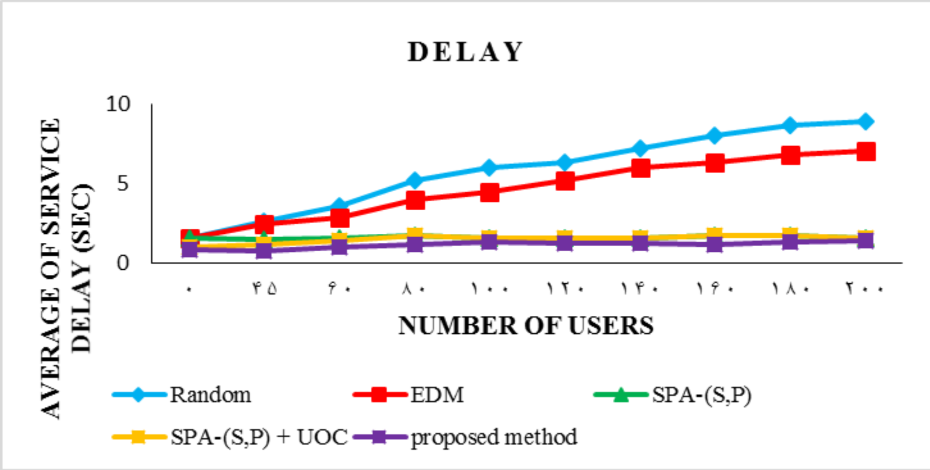


Figure 3. The average delay of primary to final services of the proposed approach in comparison with [8]

As shown in figure 3, the random curve has the highest average delay (8.9 seconds), which is much higher than other methods. In addition, EDM performs worse (7 seconds) in comparison to the proposed method. For the other two curves, the SPA results with UOC are slightly better than SPA- (S, P) (1.6 seconds) when the user number is M <150 when M> 150 is approximately equal to SPA- (S, P). However, regarding the proposed method in the present study, as well as the optimal SPA- (S, P) method with UOC strategy, the service delay results were 1.4 seconds less than other methods. These results suggest that UOC can improve user performance while ensuring network stability with less progress when the number of users is close or equal to network capacity (M = 160).

The findings are consistent with those of [8] indicating that the SPA- (S, P) algorithm, along with the UOC strategy, has the lowest delay in providing services to users. Network capacity refers to the maximum number of users that SPs can accommodate without any incompatible users. The reason why UOC can improve user performance is the existence of user switching rules designed for UOC. This switch will be allowed only if it is useful to both users and does not harm the performance of other users. From the viewpoint of SP, compulsory productivity is achieved by matching users.

6.2. SP Profit

The SP profit chart in comparison to [8] is shown in figure 4.

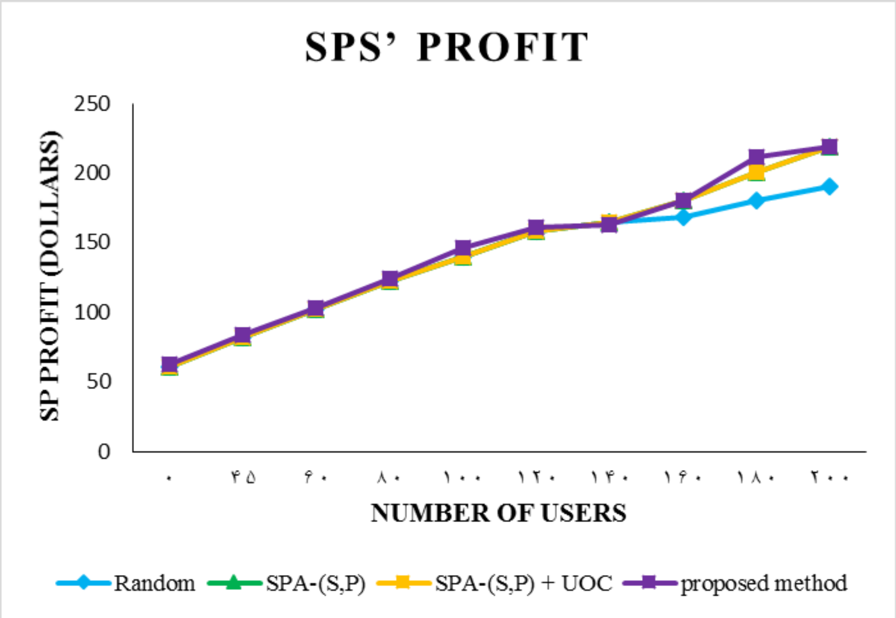


Figure 4. Primary to final SP profit of the proposed approach compared to [8]

As shown in figure 4, the average yield obtained by SPs is the same for all four methods when M <150 is the same, and the productivity is lower than other methods ($ 219) in the random method ($ 190). However, all three methods SPA- (S, P) and SPA- (S, P) with UOC as well as the proposed method are equal and better than the random method after M > 150. The productivity of SPs is measured by the number of users who are matching and those who are not. Almost all users can use different methods to adapt to a pair of resources before the number of the users reaches the network capacity. Thus, SPs can still gain all of the money. However, the users have to compete for a share when they exceed the network capacity. To determine which users should be fired or remain, users who need to be more precise offer higher prices are more probably to be selected by SPs. Meanwhile, users with higher offers make SPs more productive, which is why all three curves match SPA- (S, P) and SPA- (S, P) with UOC, and the proposed method get the same productivity when M > 150. These results are in line with [8], in which productivity was evaluated by three methods of SPA- (S, P) and SPA- (S, P) with UOC and the efficiency of the system increased by increasing the number of the users.

6.3 Cost Performance

The performance of the system costs in comparison with [8] is shown in figure 5.

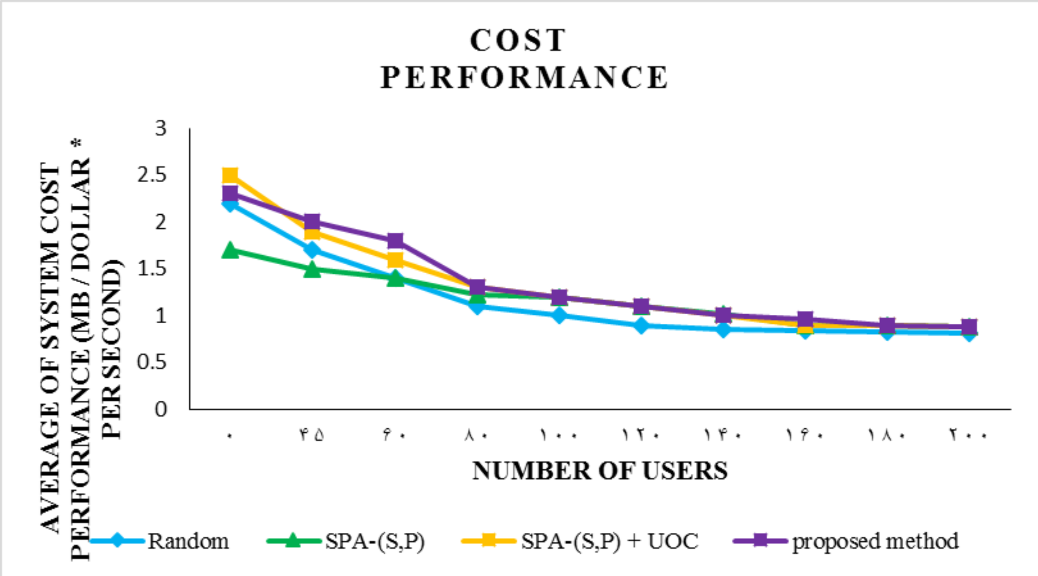


Figure 5. Cost performance of the primary to the final system of the proposed approach in comparison with [8]

As shown in figure 5, the efficiency of the proposed system cost is evaluated in four ways. Cost-performance is a measure of how much a service can be purchased, which is a joint review of the benefits of users and SPs for the purpose of allocating the best resources to users who want more (such as those with the highest prices). As shown in Figs. 2-6, when M <75, three SPA- (S, P), SPA- (S, P) curves with UOC as well as the proposed method yield better results (0.88 MB / dollar per second). However, the allocation of random sources of efficiency cost less (0.82 Mb / dollar * sec) than these three methods since users first offer their favourite resources so that some good resources may receive more offers than other sources in SPA- (S, P). Therefore, good resources which match users with their full capacity may not be as good as their spare resources when the number of users is relatively small and there are enough spare resources. On the other hand, the random allocation method is designed as a uniform random allocation in simulating this study, which allocates more users than the distributed SPA- (S, P). Therefore, all three matching algorithms are better than the random allocation method when the number of users is M> 80, which is acceptable and logical. More users lead to a smaller share of resources for each and reduce the average cost-effectiveness since the offers remain unchanged.

These results are consistent with and similar to [8] comparing the efficiency of system cost with three random methods including SPA- (S, P), SPA- (S, P), and UOC. It should be noted that the parametric effects of cache memory and CPU cycle determine the power consumption of the processor based on the execution time of the processor called IPC.

6.4. Cache performance and CPU Effects

Cache memory and CPU processor cycle were evaluated to investigate the effect of the IPC parameter on the allocation of radio and computational resources. In order to compare random, EDM, SPA- (S, P) and SPA- (S, P) + UOC, the charts and data related to the results of this research are listed separately. Based on the results in the present study (figure7), the effect of memory cache parameter on resource allocation to IoT users of IPC-based for two different tasks uses a considerable amount of the cache memory, which increases with the volume of tasks and the packages. The use of aging parameters also consumes more cache.

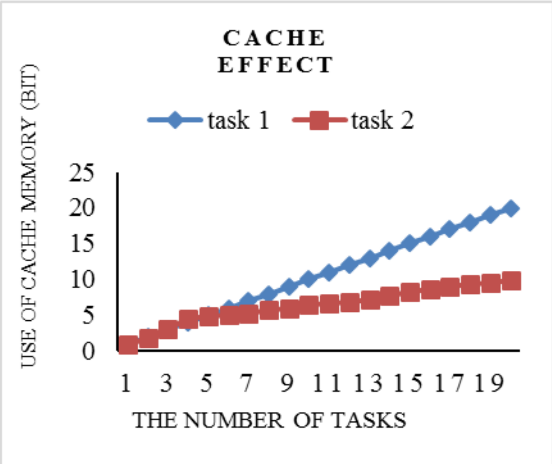
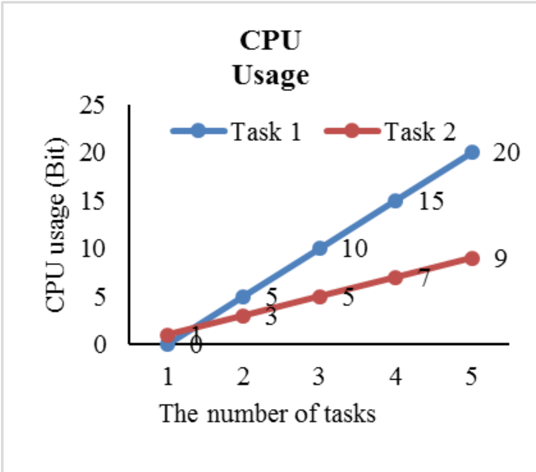


Figure 6. The effect of cache memory parameter on resource allocation to IoT users



Figue 7. The effect of CPU cycle parameter on resource allocation to IoT users

As displayed in figure 6, two different tasks use a considerable amount of cache memory regarding the analysis of the results of memory cache parameter effects on the allocation of resources to IoT users. In these two tasks, the cache increases as the number of tasks increases. The x-axis, the number of tasks, and the y-axis show the amount of cache memory. Tasks use different amounts of cache based on volume of data sent. Further, more cache is consumed by increasing the volume of tasks and the volume of its packages. As shown in figure 6, the increase in cache usage occurs by increasing the number of tasks, as well as considering the wear parameters and volume of packets, which also affect the CPU cycle in resource allocation and radio operations for the users of the IoT environment., regarding Task 1, 20 tasks with a specific volume and considering the wear parameter, use about 20 units of cache memory, which is five cycles of the processor in the range of 0 to 20 Hz. Furthermore, task 2 uses about 30 units of cache, 20 tasks, and employs a processor in 5 cycles of the processor in the range of 1 to 9 Hz. A closer look at these two sections reveals that an increase in cache usage results in reducing power consumption or CPU consumption in a given cycle, as illustrated in Task 2.

Based on the results of this study (figure 7), the effect of the CPU cycle parameter on resource allocation to IPC-based IoT users for the same two tasks used reduces energy consumption and CPU utilization efficiency. The cache usage increases when the number of tasks and the aging parameters of the packets increase, which affects the CPU cycle in the allocation of radio and computing resources to users of the IoT environment at fog level. To sum up, increasing the use of cache memory is associated with reduced energy or CPU consumption in a specific cycle

6.5 Cache performance and IPC Effects

As shown in figure 8, the result affects the CPU cycle in resource allocation and radio operations in IoT users at the fog level so that about 20 units of cache memory can be used with 20 tasks with a specified volume and considering the aging parameter. The deal is completed in 5 cycles of the processor in the range of 0 to 20 Hz from the processor. In another assessment, 20 tasks use about 30 cache caches and the processor in 5 cycles of the processor in the range of 1 to 9 Hz. In fact, increasing cache usage reduces energy or CPU consumption in a given cycle. Therefore, the cache parameter leads to an increase in computational speed in IoT-based fog processing.

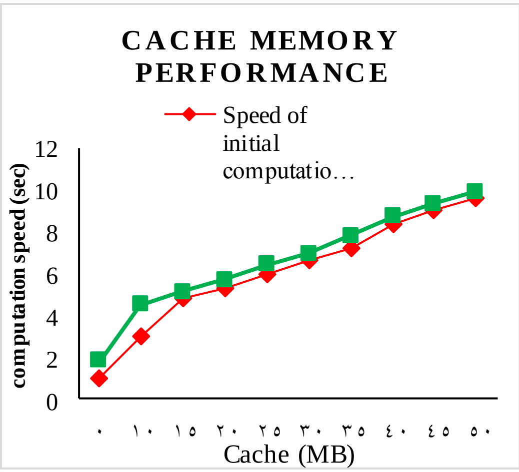


Figure 8. The average performance of the initial to the final Cache memory

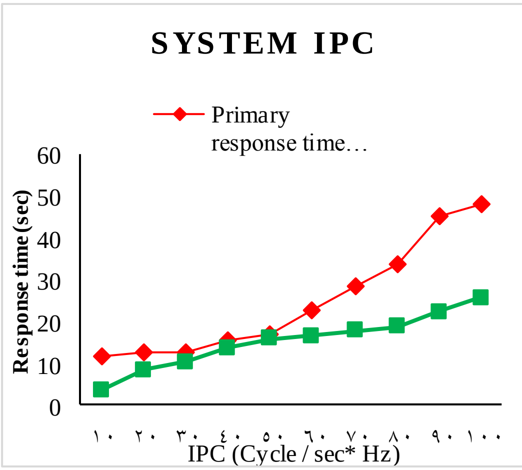


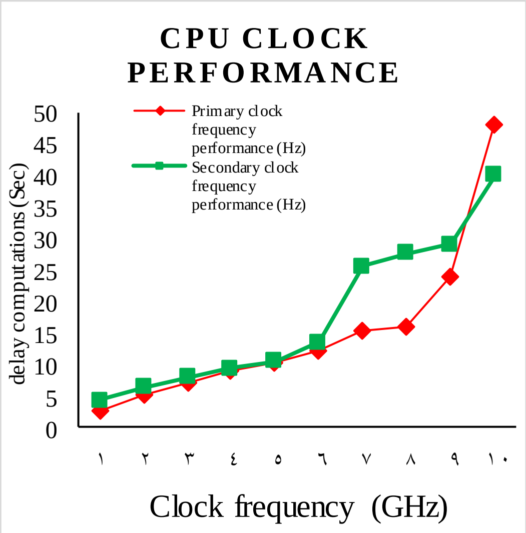
Figure 9. The average of primary to final system IPC

As displayed in figure 9, the response time for the ten tasks of the initial time balance is equal to 11.48 seconds, which was 3.51 seconds after conducting this study. Moreover, for the 100 tasks of the initial time, the load balance is equal to 48.05 seconds, which was equal to 25.61 seconds after performing this study. In addition, the results show that the cost-efficiency of the proposed approach is initially low and then expands to IoT and may sometimes increase after several stages of applications and activities or tasks. However, finally, there is a relative improvement.

The results show that the average service delay of the proposed approach with 200 users is lower than the previous methods. The SP efficiency chart, compared to the methods in [8], has almost the same values in terms of cost per dollar. The system cost efficiency chart is equal to Mb / $ * sec compared to the methods in [6].the IPC parameter accelerates the resource allocation in IoT-based fog processing.

6.6 CPU clock frequency performance and Bandwidth Effects

As displayed in figure 10, a decrease is observed in energy consumption or CPU in the CPU cycle by increasing the use of cache memory. In other words, the CPU clock frequency performance accelerates computations. Further, in the proposed model of this research, the delay rate is much lower than other methods (1.4 seconds), which indicates the effect of CPU clock frequency performance on reducing the delay in IoT-based fog processing.



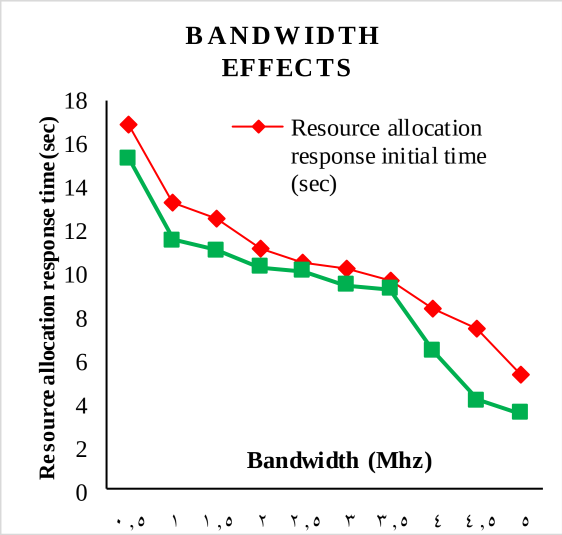


Figure 10. The average performance of the primary to final clock CPU frequency

Figure 11. Average of primary to final bandwidth

The result is showing that CPU performance is controlled by CPU cycle frequency and the CPU clock speed is optimal to reduce the executive delays at higher CPU power consumption costs. Furthermore, the findings are consistent focusing on CPU clock frequency performance and its effect on the allocation of computational resources and computational delay.The speed of resource allocation in IoT-based fog processing increases as bandwidth increases.As shown in figure 11,with a bandwidth of 1 MHz, resource allocation at response time is equal to 11.48 seconds, and increasing bandwidth to 5 MHz leads to a faster (response time3.51 seconds)

7. Conclusion

The aim of this study was to evaluate the allocation of common radio and computer resources in fog-based processing on the Internet of Things. In this research, the optimal combined method of SPA- (S, P) and UOC is used in which the effect ipc parameter on the response time in allocating radio and computing resources to IoT users is considered.

Based on the results, clock frequency performance CPU reduces IoT-based processing latency, increasing cache usage also reduces CPU / CPU power consumption in a given cycle, and increases IoT-based processing speed. Bandwidth also has a direct effect on increasing resource allocation speeds and reducing response time in Internet of Things-based cloud computing.

Reducing latency also improves response time and user access. The optimized algorithm proposed in this paper leads to a consistent consistency in the optimal allocation of resources to the system and users according to the default parameters based on server performance, system cost efficiency and service latency.

References

[1] H. Kord, O. Pourgalehdari " ALQARM: An Ant-Based Load and QoS Aware Routing Mechanism for IoT" Journal of Advances in Computer Research Sari Branch, Islamic Azad University ,Vol. 10, No. 3, pp: 65-82, August 2019.

[2] C. Iwendi; Z.Jalil ;A. R. Javed; T. Reddy G.; R. Kaluri and G. Srivastava "Key Split Watermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attacks", IEEE Access Vol 8 pp 72650 – 72660 ,15 April 2020.

[3] G. T. Reddy; M. Praveen K. Reddy; K. Lakshmanna; R. Kaluri and D. S.Rajput . "Analysis of Dimensionality Reduction Techniques on Big Data", IEEE Access Vol:8, pp54776 – 54788 ,16 March 2020.

[4] Z .Tavakoli, and; M. Fazeli,"A Study of Game Theory Methods for Improving Resource Allocation in Cloud Computing", International Conference on New Research Findings in Electrical Engineering and Computer Science, Tehran, Nikan Higher Education, 2015.

[5] A. Najafizadeh A. Salajegheh A. M. Rahmani A. Sahafi "Task scheduling in fog computing a survey " Journal of Advances in Computer Research Sari Branch, Islamic Azad University Vol. 11, No. 1, Pages: 33-5, February 2020

[6] M. K. Sohrabi and E. Khorramian "An Efficient Protocol for data Aggregation in Wireless Sensor Network "Journal of Advances in Computer Research Sari Branch, Islamic Azad University ,Vol. 6, No. 4, pp: 53-67, November2015.

[7] P. K. D.Pramanik S. Pal A. Brahmachari and Choudhury." Processing IoT Data: From Cloud to Fog-It's Time to Be Down to Earth." 10.4018/978-1-5225-4044-1.ch007.", 2018.

[8] Y. Gu, Z. Chang, M. Pan, L. Song and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing." in IEEE Transactions on Vehicular Technology, vol. 67, no. 8, pp. 7475-7484, Aug. 2018.

[9] Zeng, D., L .Gu, S. Guo, Z. Cheng, and S. Yu. Joint optimization of task s cheduling and image placement i n fog computing s upported s oftware-defined embedded system. IEEETransactions on ComputersPP, (222016.

[10] M. Taneja and A. Davya, "Resource aware placement of data analytics platform in fog computing", Procedia Computer Science 27 pp153 – 156 , 2016.

[11] Lewis G, et al. "Tactical cloudlets: moving cloud computing to the edge", In: IEEE military communications conference (MILCOM); 2014.

[12] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. Richard Yu, and Z. Han, "Computing resource allocation in three-tier IoT Fog networks: A joint optimization approach combining stackelberg game and matching", IEEE Internet of Things Journal, vol:4, no. 5, pp. 2014.

[13] Yousefpour, Ashkan, Genya Ishigaki, and Jason P. Jue. "Fog computing: Towards minimizing delay in the internet of things",In Edge Computing (EDGE), 2017 IEEE International Conference on, pp. 17-24, 2017.

[14] R. Deng, R. Lu, C. Lai, Tom H. Luan, Member, IEEE, and Hao Liang, "Optimal Workload Allocation in Fog Cloud Computing Towards Balanced Delay and Power Consumption", IEEE INTERNET OF THINGS JOURNAL, 2016.

[15] Dinh, N. T., and Kim, Y."An Efficient Availability Guaranteed Deployment Scheme for IoT Service Chains over Fog-Core Cloud Networks Sensors", (Basel, Switzerland), (2018).

[16] S.Bi, Li. Huang, Y.Jun and A. Zhang, "Joint Cache Placement and Bandwidth Allocation for FDMA-based Mobile Edge ComputingSystems",IEEE International Conference on Communications ,2020.

[17] XI LI, Y. LIU, H. JI1, H. ZHANG1,AND V. C. M. LEUNG "Optimizing Resources Allocation for Fog Computing-based Internet of Things ", Networks IEEE Access , Vol:7 , pp. 64907 – 64922 ,2019

[18] A. Ghonoodi " A Green Energy-aware task scheduling using the DVFS technique in Cloud Computing "Journal of Advances in Computer Research Sari Branch, Islamic Azad University Vol. 10, No. 1, pp: 1-10, February 2019.

[19]Y.MA, H.WANG, J.XIONG, J.DIAO and D.MA, "Joint Allocation on Communication and Computing Resources for Fog Radio Access Networks", IEEE Access , Vol: 8, pp. 108310 - 108323 ,2020.

[20] X.Huang, W.Fan, Q.Chen and J.Zhang."Energy-Efficient Resource Allocation in Fog Computing Networks with the Candidate Mechanism", IEEE Internet of Things Journal ,Vol: 7,PP 8502 – 8512, 2020.

[21] L. Liu, Z. Chang, X. Guo, S. Mao and T. Ristaniemi, "Multi-objective Optimization for Computation Offloading in Fog Computing", IEEE Internet of Things Journal, vol:5, no. 1, pp. 283-294, Feb. 2018.

[22] Zh.Huaqing , X.Yong , B.Shengrong , N.Dusit , Y.Richard and H.Zhu, "Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching", IEEE Internet of Things Journal,PP- 1204 - 1215 march 2017.

[23] H. A. El-Atta and M. I. Moussa, "Student project allocation with preference lists over (student, project) pairs",in Second International Conference on Computer and Electrical Engineering, Dubai, Dec. 2009.

[24] P.Kumar, R.Maddikunta,Th.Reddy Gadekallu, R.Kaluri, G.Srivastava, R.M.Pariz, and M.S.Khan, "Green communication in IoT networks using a hybrid optimization algorithm" Computer Communications pp159 ,May 2020.

[25] V.Kumar Billa, K.Lakshmanna, K.Rajesh, M.Praveen Kumar Reddy, G.Nagaraja and K.Sudheer. "Efficient frequent pattern mining algorithm based on node sets in cloud computing environment", Journal of Computer Science ,agu2017.

[26] G. Thippa Reddy, K. Sudheer , K.Rajesh, and K. Lakshmanna . "Employing data mining on highly secured private clouds for implementing a security-asa- service framework", Journal of Theoretical and Applied Information Technology 59(2) pp:317-326, January 2014.

[27] R.Kaluri, D.Singh Rajput, Q.Xin, K.Lakshmanna, S.Bhattacharya, Th.Reddy Gadekallu, P.Kumar and R. Maddikunta. " Roughsets-based Approach for Predicting Battery Life in IoT", Intelligent Automation & Soft Computing , Vol.27, No.2, pp:453-469, January2021.