I. J. Computer Network and Information Security, **2022, 3, 75-87**
Published Online June 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2022.03.06

Clustering Method of Mobile Cloud Computing According to Technical Characteristics of Cloudlets

Rashid G. Alakberov
Institute of Information Technology, Baku, AZ1141, Baku, Azerbaijan
E-mail: rashid.alakberov@gmail.com

Received: 20 November 2021; Accepted: 03 February 2022; Published: 08 June 2022

Abstract: The rapid increase in the number of mobile phones and IoT devices connected to the network reduces the bandwidth of the Internet communication channel, and as a result, delays occur in the delivery of data processed in remote clouds. Edge computing systems (cloudlet, fog computing, etc.) are used to eliminate resource shortages, energy consumption, and communication channel delays in mobile devices. Edge computing systems place processing devices (computers) close to users. Cloudlet-based mobile cloud computing is widely used to reduce delays in communication channels and energy consumption in mobile devices. Selection of the most suitable cloudlet allowing users to run applications fast in cloud is still a considerable problem. This paper proposes a strategy for the selection of high-performance cloudlets providing fast solutions, considering the complexity of application (file type). It offers a method for cloudlet selection out of large number of cloudlets with different technical capabilities providing faster processing of user application. The timing of user applications in cloudlets with different technical capabilities (operating frequency, number of cores, volume of RAM, etc.) also varies. The proposed method provides faster solution for the user application. User applications are grouped by type of application, and a set of cloudlets are clustered by the number of groups. Clustering is performed first by the parameters corresponding to the operating frequency of the cloudlets, then by the number of cores and the volume of RAM. The proposed method reduces energy consumption of mobile devices by providing faster processing of applications. Thus, the proposed strategy provides an energy consumption reduction on mobile devices, faster processing of results and decrease of network delays.

Index Terms: Mobile cloud computing, edge computing, cloudlet, power consumption, communication channel, file types, network delays, communication channel bandwidth, cluster.

1. Introduction

Mobile phones have become an irreplaceable part of our daily lives. With the development of mobile computing infrastructure, the number of mobile phone users has also increased intensely. People use mobile devices to prepare documents, watch movies, make online purchases, play games, use social networks, email services, browse websites, and so on. Mobile phone users can also get the Internet access anytime, and anywhere.

Mobile devices have limited computing, memory, and power resources, which hinders implementing complex software systems that require large computing and memory resources. Recently used applications by mobile users require large computing and memory resources. Performing such tasks on mobile devices cause some problems. The centralized mobile cloud computing services are widely used to solve these problems. Mobile cloud computing is a network infrastructure that processes and stores mobile users' data outside (cloud servers) the mobile devices (smartphones). Cloud technologies provide mobile users with sufficient computing and memory resources by running the tasks on a cloud platform (cloud servers). When tasks are solved on remote cloud servers of centralized cloud computing, there occur delays in data processing and delivery to the user as a result of network overload (due to reduced network bandwidth). Simultaneously, when tasks are resolved on a remote server, the power resources of mobile phones are rapidly depleted due to network delays.

Recently, due to the rapid increase in the number of mobile users using the network, the remoteness of cloud servers and loading of the Internet network there occur delays in delivering the processed data to the user. Edge computing systems (cloudlet, fog computing, edge computing, etc.) located close to the user of computing cloud resources are widely used to eliminate the mentioned problems (rapid data processing, lack of resources, power consumption, and delays in communication channels, etc.). Cloudlet computing systems are used to overcome the limitations of mobile phones related to computing and memory resources, and to reduce network latency and power consumption in mobile phones. Cloudlet (small computing clouds) is a set of devices (servers, laptops, tablets, etc.)

connected to the base stations of the mobile computing network, which provides faster delivery of information to the user than the remote cloud servers. Cloudlet network allows mobile users to use the resources of cloudlets connected to nearby base stations via Wi-Fi or 3G/4G communication channels. So, the task execution time is shorter than that of remote cloud servers. Moreover, as in computing clouds, the technical capabilities of cloudlets are higher than those of mobile devices, and this allows software applications to be processed in cloudlets that cannot be processed on a mobile device. Thus, the creation of cloudlet-based mobile cloud computing to solve these problems is a relevant issue.

2. Challenges and Goal of the Work

According to Global No.1 Business Data Platform, the number of smartphone users is estimated to reach 3.8 billion in 2021, and their monthly Internet traffic to be 12 GB/month in 2022 [1]. Demonstrated figures will cause problems in the provision of efficient cloud services to users as a result of overloaded network of centralized cloud computing. The placement of cloud computing resources close to the user is necessary to overcome this shortcoming. The user’s software supplements (tasks) are uploaded from remote cloud servers to computers located close to the user, which solves these problems rapidly. Recently, cloudlet networks based on mobile networks have been widely used for rapid data processing. Cloudlet computing networks is a cloud computing model that performs data processing and application execution not in the cloud, but in computing nodes at the network edge (as close as possible to the user). Cloudlet and cloud computing operates in an integrated manner to each other. Cloudlet computing nodes are physically closer to users than centralized data processing centers, so they can be connected faster and process the tasks more rapidly.

The computer devices used in the formation of cloudlet networks have various technical capabilities. In existing cloudlet-based mobile computing networks, the application required by the user is accessed from the cloud servers to the cloudlet network Resource Control Center (RCC) and is redirected and executed to any cloudlet with free resources. The type of application software (complexity level) and the selection of the corresponding cloudlet are not analyzed. However, if level-complexity applications (requiring large computing and memory resources) are implemented and solved on computers with low technical capabilities, then the processing time of the task is extended, and therefore, the power consumption of the mobile device increases. For the reduction of energy consumption, high-complexity applications should be implemented in cloud with high technical capabilities, which reduces the time to solve the task. First, the type (complexity level) of the user application is determined and a corresponding cloudlet with high technical capabilities is selected. Problems to be solved in the article:

- to reduce energy consumption of mobile devices.
- to provide fast processing of tasks.
- to select cloudlets according to the type of application.

Thus, the article proposes a strategy (mathematical model) to solve the above problems:

- Grouping the applications according to types (degree of complexity).
- Developing an algorithm that ensures the sequence arrangement of cloudlets.
- Clustering the cloudlets according to technical capabilities.

The proposed strategy can reduce the energy consumption of mobile devices and delays in application processing by uploading applications to cloudlets according to the complexity level.

3. Related Work

In some studies, edge computing between mobile users and cloud servers have been analyzed to reduce energy consumption and network delays in mobile devices [2-4]. In a cloud-based mobile computing system, the efficient distribution issue of mobile users’ tasks among multiple cloudlets was considered [5]. The paper [6] proposes the joint optimization method for network delays and energy efficiency in mobile devices. The studies [7-10] focus on the usage of mobile cloud computing to solve problems (resource scarcity and power consumption, network latency, etc.) that require large computing and memory resources on mobile devices. Some studies demonstrate that mobile users’ software reduces energy consumption on mobile devices by migrating them from cloud servers to cloudlets [11-12]. The paper [13] collects and analyzes the latest power-based offload protocols and architectures, as well as scheduling and balancing algorithms used for cloud green computing. This review compares system architectures by identifying their most notable advantages and disadvantages. Existing support structures are classified and compared based on the stage of the task offloading. The paper [14] analyzes the delays in the delivery of results to users in the solution of the problems on cloud servers. The usage of cloudlets is recommended to eliminate delays. Several studies [15-17] show that the usage of edge computing systems reduces the energy consumption of the user device. This paper [18] analyzes how to protect user data in mobile cloud. In terms of objectives, this study is a comprehensive analysis of existing methods that can be used to effectively protect data in MCC. In addition, this work will provide an advanced roadmap for the research and development communities to make the right choice of the proposed approach. An energy-efficient scheduling strategy for the cloud is proposed in [19], where the virtual machines (VMs) are allocated, migrated, and canceled dynamically to make the cloud scalable. Some studies explore the user's location to improve service quality and reduce costs and focus on the placement tasks of mobile users in cloudlets based on location [20]. The paper [21] explores the issues of energy consumption reduction in mobile devices using cloudlets. In [22], the balanced distribution issue of resources in the cloudlet network is analyzed, considering the users' mobility, and proposes an algorithm. This paper [23] addresses security issue by proposing usage of obfuscation and encryption techniques to scramble the data at IoT devices which is later stored in encrypted form at cloud server. The work [24] presents the development of a method for choosing a cloud computing platform for serving government agencies, which will form the digital government in the future. The paper [25] proposes mathematical models of the process of information interaction between a set of subjects (IoT devices) and a base station in the 5G-IoT ecosystem. The studied process, in this case, is represented by a queuing system with a flow of new incoming requests with needs for the desired volumes of system resources and a flow of service signals, the receipt of which initiates redefinition of allocated for received incoming requests volumes of system resources. In [26], a multi-layered computing model was proposed to increase the productivity of mobile devices using mobile cloud computing systems. The paper [27] analyzes the reduction of energy consumption and delays in solving problems by the usage of Mobile Edge Computing (MEC) technologies at computing nodes located near mobile devices. In several studies [28], the creation of cloudlets by mobile users near base stations is considered. In [29], a cloudlet allocation method is proposed for improving QoS of the system. The paper [30] discusses the placement of user software applications in nearby cloudlets. The paper [31] proposes an algorithm for the selection of a cloudlet, providing optimal distribution of problems between cloudlets and rapid solution of problems. The study [32] analyzes the reduction problems of the energy consumption of mobile devices and elimination of delays in communication channels by solving complex computing and memory resource-intensive tasks in the cloud nearby the user. The study [33] proposes the development of algorithms for sharing the resources of cloud servers. The paper [34] explores the problem of reducing delays by reducing the number of communication channels and ensuring that users' tasks are solved in the cloudlets closest to them. A dynamic offloading strategy for cloudlet is proposed in [35] for reducing the energy consumption. [36] considers the selection problems of the most convenient cloudlet out of the numerous available cloudlets according to the type of mobile user application. The paper [37] considers the creation of a cloudlet network infrastructure in the Wireless Metropolitan Area Network environment and highlights the prediction of deployment of cloudlets near certain base stations.

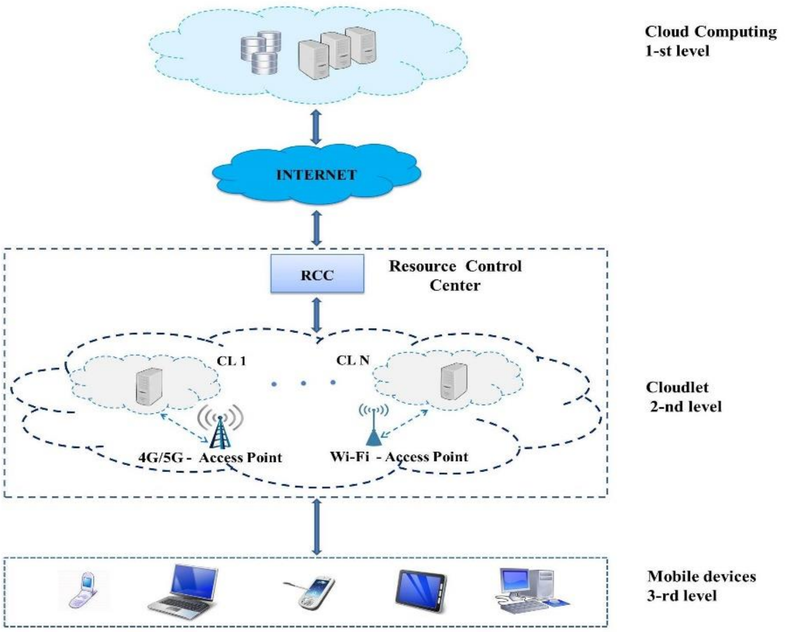

Fig.1. Cloudlet-based mobile cloud computing architecture

4. Cloudlet-based Mobile Cloud Computing Architecture

The architecture of hierarchically structured cloudlet-based mobile cloud computing is shown in Figure 1 [9, 30]. Components of Cloudlet-based mobile computing networks are mobile phones, base stations (4G / 5G, Wi-Fi), cloudlet networks, Internet networks, cloud computing. Level 1 contains the servers of the cloud computing system, Level 2 is called the mobile network layer and consists of mobile operators, base stations, connected cloudlets, and the Resource

Control Center. Level 3 consists of heterogeneous mobile phones (smartphones). According to the mobile user request, the software is called to the cloudlet network from remote servers via RCC and its solution is implemented by downloading to the appropriate cloudlets. RCC provides mobile users with services such as home agent (HA) stored in the database and AAA (authentication, authorization, and accounting) based on subscriber information. RCC contains information about the location of cloudlets and their technical capabilities. The data identifiers (IDs) of cloudlets and mobile users are stored in the RCC in the data table form, and redirects applications to vacant cloudlets in the network according to the user request, resolves and communicates the results to the user. RCC collects information about the technical characteristics of the computer equipment used to create cloudlets (processor frequency, number of cores, number of virtual machines, memory capacity, etc.) and about which cloudlet is located near the mobile network users. The control server located in the RCC on the basis of the data table determines the cloudlet where the problem of the mobile user is solved. The central server in RCC also provides inter-cloud or user-cloud data exchange.

Simultaneously, the control center contains information about the computer equipment used to create cloudlets. Thus, mobile network users download and resolve applications to the cloudlets closest to them, which in turn release the network from overloading. This architecture can partially solve some of the mentioned problems (problem-solving time, power consumption, delays in the communication channel, interruptions, etc.). Advantages of cloudlets include rapid access to the service, support for mobility, reduction of roaming costs. Thus, considering the complexity of the problem, the selection of the appropriate cloudlet and the problem-solving time provide energy savings for mobile devices and reduce network delays. Therefore, according to the received request, the placement of the software applications called from the cloud servers in the cloudlet providing the user's requirements is important.

5. The Method for the Selection of Cloudlets According to the Types of Applications

The paper focuses on a solution problem of efficient use of cloudlets resources located near the base stations of Wireless Metropolitan Area Networks (WMAN). Considering the size of WMANs, incorrect placement of user tasks in the cloud can lead to processing delays and there occur delays in delivering the results to the user. We should place cloud computing resources close to the user to overcome this shortcoming. In this case, cloudlets process requests rapidly and allow mobile devices to use less power.

Before choosing a cloud, you need to inspect the network and determine the status. Thus, after the inspection in the RCC, we can find information about the technical characteristics of the cloudlets, and when the request is received, we can determine in which cloudlet the problem is solved rapidly.

Potential cloudlets should be evaluated by RCC when deciding whether to place the software tool in any cloudlet. Some criteria (high technical capabilities, availability of free resources, etc.) should be considered in the selection of the cloudlet during the evaluation.

To find the most suitable application from the cloudlets and download it is still a substantial problem for the user. The study proposes the selection of a cloudlet providing rapid processing of the user application from numerous cloudlets with different technical capabilities. The problem-solving time of user applications in cloudlets with different technical capabilities also varies. Therefore, the selection of the most convenient cloudlet among the various cloudlets according to the type of application used by the user is recommended. Using the proposed method, we ensure a reduction in the energy consumption of mobile devices by providing rapid implementation of programs.

Thus, the accurate selection of cloudlets according to the type of applications used by users eliminates the above-mentioned problems. Due to the type of application to be used, the cloud selection method which solves the problem rapidly among other cloudlets reduces energy consumption on mobile devices, network delays. For instance, the types of applications to be used (downloaded) can be classified as follows [36]:

1. Application software group (AS1) requiring large computing and memory resources: algebra, numerical, operational research, graph theory, translate, cryptography, etc.
2. Application software group (AS2) requiring average computing and memory resources: game, web service, health, banking services, etc.
3. Application software group (AS3) requiring low computing and memory resources: file creation, sorting, searching, file conversion, etc.

A cloudlet-based mobile computing network is composed of numerous cloudlets based on non-homogeneous computers (servers, desktops, notebooks, etc.). Information on the technical capabilities of the computer equipment used in the creation of cloudlets is placed in the form of a table in the RCC. The cloudlets set are also grouped according to the number of groups of applications used. In our case, we divide cloudlets into three groups. The computer equipment included in the *C1* cloudlet set has high technical capabilities. The computer equipment included in the *C2* cloudlet set has average technical capabilities. Computer equipment included in the *C3* cloudlet set has low technical capabilities.

- *C1* cloudlet set - is based on computers with technical capabilities in the following figure range: (CPU: 2.2 -

3.5 GHz, number of cores: 12-24, RAM: 16-32 GB)

- *C2* cloudlet set - is based on computers with technical capabilities in the following numbers range: (CPU: 1.8-2.4 GHz, number of cores: 8-12, RAM: 6-12 GB, etc.)
- *C3* cloudlet cluster is created on the basis of computers with technical capabilities in the following numbers range: (CPU: 1.2-2.0 GHz, number of cores: in the range of 1-6, RAM: in the range of 2-6 GB)

Thus, we group users' software according to the degree of complexity, and cloudlets according to their technical characteristics. Considering the request of the user, RIM selects the cloudlet according to the type of its application. Cloudlet network infrastructure is dynamic. That is, new cloudlets can be added to this structure or disconnected from the network. Therefore, to the arrangement of the cloudlets in the RCC according to certain parameters (computing power, frequency, number of cores, etc.) and the selection of a cloudlet from set of cloudlets in the sequence alignment of the software takes a long time. Because the addition or opening of a new cloudlet requires re-creating the sequence alignment. Due to the proposed method, clusters of cloudlets are created according to the technical characteristics, the user's request is directed to the appropriate cluster, and the selection time of the cloudlet is reduced several times.

If there doesn’t exist any vacant virtual machines in the C1 cloudlet, the task is directed to the C2 cloudlet set. In general, if there doesn’t exist any vacant resources in the cloudlet sets, the solution is kept in standby mode by RCC. When vacant resources appear in the appropriate cloudlet set, the task is redirected to it. Thus, the selection of the appropriate cloudlet in solving the user’s task reduces the solution time, energy consumption, delays. Considering the above-mentioned, the following model has been proposed to reduce energy consumption and network delays in mobile devices.

The selection of one of the thousands of cloudlets that provide rapid processing of the user’s software (task) and provide a solution is a great problem. The selection process is time-consuming and may not provide a rapid solution to the task in the selected cloudlet. If the cloudlet to be selected is chosen from N different cloudlets, then these cloudlets should be arranged in the RCC according to the computing power, frequency, number of cores, memory size, etc. The sequence alignment can simplify the search for the necessary cloudlet. Let’s look at the algorithm that ensures the sequence alignment of cloudlets.

Let's enter the following indices:

N- the total number of cloudlets in a sequence alignment;
M- the number of comparison operations used in the selection of cloudlets;
R_i – the computing power (frequency) of the *i*-th cloudlet;
C_j – *j*-th cloudlet;
Z- backup address to store temporarily removed cloudlets;
i, j, k, l - are the indices used in the algorithm.

If there we have *n* cloudlets, the following algorithm (Sequence Alignment Algorithms) can be used to obtain their sequence alignment:

Step_1: Start;
Step _2: i=2;
Step _3: j=1;
Step _4: If $R_i<R_j$, go to step 5, otherwise go to step 9;
Step _5: j=j+1;
Step _6: If j<i, go to step 4, otherwise go to step 7;
Step _7: i=i+1;
Step _8: If i<N, go to step 3, otherwise go to end;
Step _16: End;
Step _9: $Z=C_j$;
Step _10: k=i-j;
Step _11: l=1;
Step _12: $C_{i-(l-1)}=C_{i-l}$;
Addım_13: l=l+1;
Step _14: If l<k, you should go to step 12, otherwise go to step 15
Step _15: C_i=Z, go to step 7:

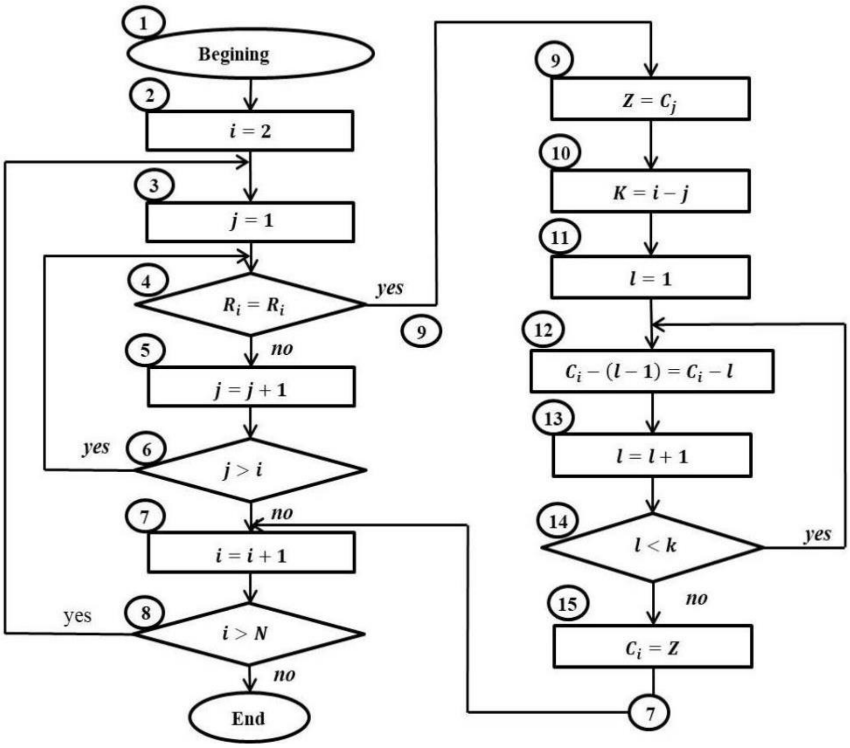

Fig.2. Block diagram of the Sequence Alignment Algorithms of cloudlets.

This algorithm function can be stated by the following Delphi fragment (subroutine):
Procedure Tform1. Alignment;
We have

```
i, j, k, l: integer;
Label  Lab_03, Lab_04, Lab_05, Lab_07, Lab_12;
     Start
     i := 2;
 Lab_03:
     j:= 1;
 Lab_04:
     If   R [i] < R [j] then go to Lab_05;
     Z := R [j];
     Z_F:=C[j];
     k := i-j;
     L:=1;
 Lab_12:
     R[i-(l-1)]:=R[i-l];
     C[i-(l-1)]:=C[i-l];
     l:=l+1;
     If l< k then go to Lab_12;
     R[i]:=z;
     C[i]:=Z_F;
      Goto lab_07;
 Lab_05:
     j:=j+1;
     If j<i then go to Lab_07;
     Goto Lab_04;
 Lab_07:
     i:=i+1;
     if i<I then  Go to Lab_Fin;
     Goto Lab_03;
 Lab_Fin:
 End;  // end of proc.
```

Analysis of the Sequence Alignment Algorithms shows that when there are N cloudlets, should be performed $M = 1 + 2 + \cdots + (N-1) = \frac{(N-1)*N}{2}$ comparison operations. Therefore, the time selection of one of the N cloudlets can be written as a function $T(N),\ M = \frac{(N-1)*N}{2}$.

$$T(N) = F(M) = F\big(1 + 2 + \cdots + (N-1)\big) = F(\frac{(N-1)*N}{2})$$

The F function in this formula can usually be written as a linear function. Given the difficulty of searching for a large number of cloudlets, as mentioned above, it can be divided into several approximately equal parts in a certain order. In this case, less operation is required to search for a cloud from a small set. If the set of cloudlets is divided into approximately equal parts by k, the number of operations can be reduced by k squares. Thus,

$$M^k = \frac{\left(\frac{N}{k}-1\right)*\frac{N}{k}}{2} \quad \text{and}$$

$$T(M^k) = F(M^k) = F(\frac{\left(\frac{N}{k}-1\right)*\frac{N}{k}}{2}).$$

You can see how many times the number of operations decreased by the following limit.

$$\lim_{N\to\infty} \frac{\frac{(N-1)*N}{2}}{\frac{\left(\frac{N}{k}-1\right)*\frac{N}{k}}{2}} = \lim_{N\to\infty} \frac{N^2 - N}{\frac{N^2}{k^2} - \frac{N}{k}} = \lim_{N\to\infty} \frac{N^2 - N}{N^2 - kN} k^2 = \lim_{N\to\infty} \frac{N^2(1-\frac{1}{N})}{N^2(1-\frac{k}{N})} k^2 = k^2$$

If the indices $\frac{1}{N}$ and $\frac{k}{N}$ approach N infinity, the value of the limit becomes k^2 when it goes to zero.

If *N* number of cloudlets are divided into a set of cloudlets *C1*, *C2* and *C3* as shown above, the number of operations for the selection of the required cloudlet can be significantly reduced.

Suppose that the number of cloudlets included in each of the separated cloudlets is equal to one-third of N (because it is divided into three parts). It can be achieved that the number of cloudlets in the separated clusters is approximately equal to $\frac{N}{3}$. In this case, the number of comparison operations in each newly divided set should be

$$M^j = \frac{\left(\frac{N}{3}-1\right)*\frac{N}{3}}{2}$$

The maximum time for the selection of a cloudlet from this set can be written as

$$F(M^j) = F(\frac{\left(\frac{N}{3}-1\right)*\frac{N}{3}}{2}) \ .$$

$$\lim_{N\to\infty} \frac{\frac{(N-1)*N}{2}}{\frac{\left(\frac{N}{3}-1\right)*\frac{N}{3}}{2}} = \lim_{N\to\infty} \frac{N^2 - N}{\frac{N^2}{9} - \frac{N}{3}} = \lim_{N\to\infty} \frac{9(N^2 - N)}{N^2 - 3N} = \lim_{N\to\infty} \frac{9N^2(1-\frac{1}{N})}{N^2(1-\frac{3}{N})} = 9$$

Since the indices $\frac{1}{N}$ and $\frac{3}{N}$ go to zero when approaches N infinity, the value of the limit is equal to 9.

This shows that if the number of cloudlets is large enough, the number of operations performed for the selection of a cloudlet from one of the separated clusters will be nine times less after dividing the cloudlets into three equal clusters. This shows the advantage of the clustering operation.

Let's look at the following patterns to illustrate this. Let, N = 3000 is the total number of cloudlets.

For simplicity in the separated 3 (three) cloudlet clusters, the number of cloudlets is taken in the cluster equal to each other, then $\frac{N}{3} = 1000$. Then, the time to select one of the cloudlets clusters of N=3000 according to the type of application will be $M = \frac{(N-1)*N}{2} = \frac{(3000-1)3000}{2} = 4498500$ (it is necessary to perform a comparison operation). And the time to select one of the cloudlets clusters of N=1000 will be $M^j = \frac{\left(\frac{N}{3}-1\right)*\frac{N}{3}}{2} = \frac{\left(\frac{3000}{3}-1\right)*\frac{3000}{3}}{2}$= 499500 (comparison operation is required). If cloudlets are clustered as k=3, then the number of selection operations can be reduced by

$k^2 = 9$ time. Thus, $M\frac{M}{M^j} = \frac{4498500}{499500} \approx 9{,}006.$

So, presented difficulties for selection of the required cloudlet from a large number of cloudlets can be divided into several clusters according to a certain rule (according to technical characteristics). In this case, less operation is required to search for a cloud from a separated small set. In our example, as a result of clustering, the time spent on selecting the cloudlet will be reduced by 9 (nine) times.

Considering this, we should reduce the time spent searching and cluster the number of cloudlets to select the correct cloudlet.

Clustering should be carried out in two directions:

- Clustering the software by their type (format) (as mentioned above)
- Clustering the cloudlet sets based on the technical characteristics of the computers involved in creating cloudlets.

If you solve the software in cloudlets with different technical capabilities, the high-tech cloudlet solves the problem rapidly. Let's look at the clustering issue according to different technical characteristics of cloudlets:

If, x and y (cloudlets) are two cloudlets entering in one, two, or a combination of the sets *C1, C2* or *C3* described above, the distance between them (d-metrics) may be one of the metrics that satisfy the following properties:

1. $d(x, y) \geq 0$ non-negative property;
2. The equality $d(x, y) = 0$ is satisfied only if the cloudlets x and y have the same characteristics. The characteristics of any cloudlet create a vector. The distance from the vector to itself is zero;
3. $d(x, y) = d(y, x)$ symmetric property;
4. $d(x, y) \leq d(y, z) + d(z, x)$ triangular property.

The following metrics are used in practical applications.

- Minkowski metric

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^q \right)^{1/q} \text{ where } q \text{ is a positive integer.}$$

- Euclidean metric

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

As seen, Minkowski and Euclidean metrics are analogous to each other. If, q=2 the Minkowski metric transforms to the Euclidean metric.

- Maximum metric

$$d(x, y) = \max_{i=1}^{n} |x_i - y_i|$$

- The modified Manhattan distance is equal to the average value of the modulus of the corresponding characteristic differences.

$$d(\hat{x}, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|$$

Any of the metrics shown or not shown here can be used to cluster the cloudlet sets. However, calculations may be simpler when using the Manhattan metric. Simultaneously, as numerous parameters (frequency, number of cores, memory, etc.) are used to create clusters, the usage of the Manhattan metric would be more appropriate. So, if a characteristic of cloudlets is used, the metrics can be written as follows:

$$d(x, y) = abs(x_1 - y_1) \tag{1}$$

Here, a selected characteristic of the x and y cloudlets, which calculates the distance between $x_1$and $y_1$for instance, the frequency of the CPU. If the number of nuclei is added to the formula [1],

$$d(x,y) = \frac{1}{2}\sum_{i=1}^{2} abs(x_i - y_i) \quad (2)$$

If all 3 characteristics are used, then the distance between the clouds can be calculated as the following formula (3):

$$d(x,y) = \frac{1}{3}\sum_{i=1}^{3} abs(x_i - y_i) \quad (3)$$

However, it is obvious that the selected characteristics are of different nature and prices are very various. For the elimination of this difference, the values of the characteristics should be normalized. For this purpose, the maximum value of the j-th characteristic for all sets is found. Let's call this

x_{jmax} maximum. The maximum of the *J*-th characteristics.

The value of the corresponding characteristic of $\forall\ x \in C1 \cup C2 \cup C3$ cloudlet is changed as

$x_j = x_j / x_{jmax}$.
$x_j \in [0,1]$. Then, a cluster can be formed by performing comparison operations. For the first cluster
$k1 = \emptyset$, let's take the cloudlet x which is chosen as the center of the cluster, and any d1 as the diameter of the cluster.

For $\forall\ y \in C1 \cup C2 \cup C3$ cloudlets to belong to the first cluster

$d(x,y) \le d1$ condition must be satisfied. If this condition is satisfied, then $k1 = k1 \cup \{y\}$.

Thus, the first cluster is formed. Other clusters can be formed in an analogous way. After the formulation of all the necessary clusters, they can be renamed as desired. Each cluster contains cloudlets close to the cloudlet selected as the center.

Let's show a computer model of cluster formulation (let's show it with a visual pattern). Computers with the specifications given above will be used to create sets *C1, C2 and C3*. Let's normalize three technical characteristics of computers with different units of measurement (0-1 range). Normalized values (maximum frequency value - 3.3 Ghs, maximum number of cores - 24, maximum memory capacity - 32 GB) are shown in Table 1.

Table 1.

	Cluster 1		**Cluster 2**		**Cluster 3**	
Maximum values of characteristics	Technical characteristics.	Normalized.	Technical characteristics.	Normalized.	Technical characteristics.	Normalized.
CPU Max. 3,5 Ghs	2,2–3,5	0,628–1	1,8–2.4	0,514–0,685	1,2–2,0	0,342–0,571
Core number Max 24	12–24	0,5–1	8–12	0,33–0,5	1–6	0,04–0.25
RAM Max. 32 Gb	16–32	0,5–1	6–8	0,18–0,25	2–6	0,06–0,18

Let's look at how to formulate a metric d (x, y) for the formulation of a cluster *C1*: First, the processor frequency is used to create the cluster. First, we define the interval of the CPU - 0.628 -1 and then determine the center of the interval. Then a set of clusters is defined accordingly. As shown in Table 1, cloudlets with the same frequencies can be in different clusters, so the set of cloudlets (cluster C1) can be accurately determined using the second characteristic (number of cores) and then the third characteristic (memory).

If, we first determine the center (M_T) according to the first characteristic of the cluster *C1* then,

$$M_T = \frac{1 + 0{,}628}{2} = 0{,}814$$

The radius (r_T) is

$$r_T = \frac{1 - 0{,}628}{2} = 0{,}186$$

If it is intended to create a cluster using only one characteristic, cloudlets that satisfy the following inequality of the frequency characteristic can be included in the cluster:

$$abs(0{,}814 - y_1) \leq 0{,}186$$

Moreover, a cluster can be created according to the core. Let’s determine the center and radius of the cluster created by the core:

$$M_N = \frac{1 + 0{,}5}{2} = 0{,}75$$

$$r_N = \frac{1 - 0{,}5}{2} = 0{,}25$$

If it is intended to create a cluster using cores, the cloudlets that satisfy the following inequality of the core characteristic can be included in the cluster:

$$abs(0{,}75 - y_2)) \leq 0{,}25$$

Maybe the created cluster *(C1)* includes some cloudlets used in the *C 2* set (for instance, by frequency). The second characteristic (number of cores) can be used to create a cluster *(C1)* to remove the cloudlets (other characteristics that do not match) that are included in cluster *C1* by frequency.

If it is intended to create a two-dimensional cluster, the center of the cluster will be at $M(0{,}814; 0{,}75)$ point. The radius of this cluster can be calculated as $r = \frac{r_T + r_N}{2} = \frac{0{,}186+0{,}25}{2} = 0{,}218$. In this case, the cloudlets that satisfy the following inequalities of the selected characteristics are included in the cluster.

Let’s look at creating a cluster using cloudlets satisfying the condition $((abs(0{,}814 - y_1) \leq 0{,}186)\ and\ (abs\ (0{,}75 - y_2)) \leq 0{,}25)$ of the first cluster *(C1*.

A cluster created on the basis of two criteria can be formed from clusters that satisfy the condition $\frac{1}{2}(abs(0{,}814 - y_1) + abs(0{,}75 - y_2)) \leq 0{,}218$.

In this case, the obtained cluster is in the form of a rectangle ABCD (Figure 3). Thus, cloudlets on two parameters (frequency and number of cores) inside the rectangle ABCD form a set of cloudlets *C1*. A square is used to select a cloudlet with higher technical capabilities than the cloudlets in this set.

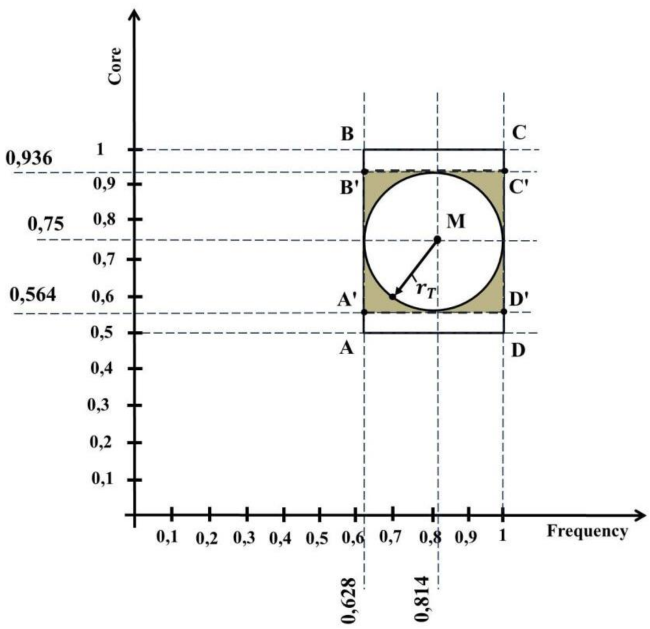

Fig.3. Two-dimensional (with r_T radius) cluster.

So, in some cases, $r = r_T = 0{,}186$ can be taken, because of $r_T < r_N$. In this case, the inequality to be satisfied will be $sqrt(\frac{1}{2}(sqr(0{,}814 - y_1) + sqr(0{,}75 - y_2))) \leq 0{,}186$.

The obtained cluster will be in the form of a square A 'B' C 'D' inside the rectangle ABCD (Figure 3). The circle radius is used to select cloudlets with higher technical characteristics from the cloudlets included in the square cluster.

Thus, cluster *C1* contains cloudlets with high technical characteristics that enter a circle with an equal radius ($r_T = 0{,}186$) (Figure 3). As observed in the figure, the user solves the problem by selecting one of the vacant cloudlets inside the circle with radius $r_T = 0{,}186$. Also, clustering can be implemented using parameter 3 (memory) to provide rapid processing of the problem in the cloudlet set. According to the three parameters, the distance between the cloudlets can be calculated as follows:

$$d(x,y) = \frac{1}{3}\sum_{i=1}^{3} abs(x_i - y_i)$$

If three characteristics are used, a sphere is obtained instead of a circle. If there doesn't exist a vacant resource in cluster *C1*, the software execution goes into standby mode or the search is performed in cluster *C2* or cluster *C3*.

Thus, by increasing the number of parameters, we can create a cluster that provides rapid processing of the application problem, and reduce the time to select the cloudlet according to the software format several times.

6. Evaluation of Experimental Results

Considering the complexity of mobile users' software applications, the comparison of problem-solving times by placing them in appropriate cloudlets was analyzed. The experiment was conducted on cloudlets with various technical capabilities created in the AzScienceNet science computer network of the Azerbaijan National Academy of Sciences. For the experiment, three computer equipment with the following technical capabilities of the clusters *(C1, C2, C3*) have been used:

- Cloudlet included in the *C1* cloudlet set: Intel Xeon E5 - 2650, 2.3 Ghs, core-16, RAM - 32 GB.
- Cloudlet included in the *C2* cloudlet set: Intel Core i7, 2.2 Ghs, core-8, RAM - 8 GB.
- Cloudlet included in the *C3* cloudlet set: ASUS Zenbook, Intel Core i5, 1.8 Ghs, core-2, RAM - 4 GB.

As the experiments were conducted within the corporate network, the uploading of the issue to the cloudlets, obtaining the results, and delays in the network were not considered. Also, the experiment considered the solution of problems included in the clustering of applications with a high degree of complexity. For the conduction of the experiment, the multiplication issues of matrices of size 10000x10000 using the packaged software MATLAB R2020b was considered.

Via the experiments, it is determined that the selection of cloudlets according to the complexity level of the user application reduces the problem-solving time. The results of the experiment are shown in Table 2.

Table 2. Problem-solving time in different cloudlets

Cloudlet	Processor	Core	RAM	Operation System	Multiplication time of matrices of size 10000 × 10000 (seconds)
C1	Intel Xeon E5-2650, 2.3 Ghz	16	32 GB	Windows 10 × 64 bit	3,1794
C2	Intel Core i7, 2.2 Ghz	8	8 GB	Windows 10 × 64 bit	11,0634
C3	Intel Core i5, 1.8 Ghz	2	4 GB	Windows 10 × 64 bit	14.0813

As seen from the table, when the matrix multiplication task is solved in cloud *C1*, the solution time will be 4.43 times less than in *C3,* which will reduce the power consumption of the mobile device in standby mode. If we increase the size of the matrix, this number will increase repeatedly. As seen from the experiment, the disorderly placement of users' problems on the cloudlet network prolongs the problem-solving time. Computers with high computing productivity solve tasks rapidly. Therefore, users' tasks should be addressed by RCC to a cloudlet with high technical capabilities, considering their level of complexity (format), so that the issue can be resolved rapidly. The sooner the task is resolved, the lower the power consumption of the mobile device in standby mode.

7. Future Research

Future research will include studies in the field of cloudlet-based mobile cloud computing. It is planned to develop methods and algorithms to solve problems such as development of a planning model for the efficient use of mobile cloud resources, pre-deployment of high-frequency software applications in the cloudlet network, optimal modeling of tasks and resources in mobile cloud computing environment, efficient and balanced use of cloudlets in cloud computing, the method of placing cloudlets close to user to eliminate delays in communication channels, etc.

8. Conclusion

The paper focuses on a method of selecting cloudlets according to the complexity level of user applications. The usage of edge computing systems - cloudlet networks is proposed to eliminate resource shortages, power consumption, and delays in communication channels on mobile devices. The architecture of cloud-based mobile cloud computing has been analyzed to reduce the energy consumption of users using mobile cloud computing services on mobile devices. Selection of the appropriate cloudlet from the arranged clodlets took (9 times) more time than the selection of the cloudlet from the clustered set. An algorithm was developed providing sequence alignment of the cloudlets within the cluster. Cloudlets were clustered according to their technical characteristics. The clustering was first performed by the parameters corresponding to the operating frequency of cloudlets, then by the number of cores and the volume of RAM. A method was proposed to select a cloudlet with high technical capabilities from the cloudlet sets according to the type of application software. The proposed method provided reduction of energy consumption in mobile devices, rapid processing of results, and reduction of network delays. Considering the complexity degree of mobile users' applications software, a comparison of their processing (solution) times in the corresponding cloudlets was analyzed. As Table 2 showed, when matrix multiplication was processed faster in cloud C1, the resolution time would be 4.43 times less than in cloud C3, which would reduce the power consumption of the mobile device in standby mode. The proposed strategy provided a reduction in energy consumption in mobile devices and fast processing of results.

Reference

[1] Global No.1 Business Data Platform:https://www.statista.com /statistics/738977/worldwide-monthly-data-traffic-per-smartphone.

[2] M.Yuyi, C. You, J. Zhang, K. Huang and K. Letaief. (2017). "A survey on mobile edge computing: The communication perspective." *IEEE Communications Surveys & Tutorials* vol. 19, no. 4, pp.2322-2358.

[3] A. Nasir, Y. Zhang, A. Taherkordi and T. Skeie. (2017). "Mobile edge computing: A survey." *IEEE Internet of Things Journal* vol. 5, no. 1 pp.: 450-465.

[4] A. Mukherjee, D, Soumya, K. Ghosh and R. Buyya. (2021). "Introduction to Mobile Edge Computing". November 2021.DOI:10.1007/978-3-030-69893-5_1.In book: Mobile Edge Computing.pp.(3-19).

[5] A. Muneera, M. Al-Ayyoub, Y. Jararweh, L. Tawalbeh and E. Benkhelifa. (2016)."Power optimization of large-scale mobile cloud system using cooperative cloudlets." *In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops* (FiCloudW), pp.34-38.

[6] M. Sachula, Y. Wang, Z. Miao and K.Sun. (2018)."Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment." *Peer-to-Peer Networking and Applications*, vol. 11, no. 3, pp. 462-472.

[7] R.P. Mathur and M. Sharma. (2019). "A survey on computational offloading in mobile cloud computing". 2019 *Fifth International Conference on Image Information Processing*, 8985893, pp.525-520.2019.https://doi.org/10.1109/ICIIP 47207.

[8] R.K Alekberov. (2021). "Strategy for reducing delays and energy consumption in cloudlet- based mobile cloud computing," *International Journal of Wireless Networks and Broadband Technologies*, vol. 10, no.1, pp. 32-44.

[9] R.S. Somula, S. Ra. (2018). "A survey on mobile cloud computing: Mobile Computing + Cloud Computing (MCC = MC + CC)." *Scalable Computing: Practice and Experience*, vol. 19, no. 4, pp. 309–337.

[10] A. Mukherjee and D. De. (2016). "Low power offloading strategy for femto-cloud mobile network". *Eng Sci Technol Int J*, vol.19, pp.260–270.

[11] R. Hassan, N. Yazdani and R. Shojaee. (2017)."Modeling and performance analysis of cloudlet in Mobile Cloud Computing." *Performance Evaluation*, vol. 107, pp. 34-53.

[12] S. Xiang and N. Ansari. (2020)."Green cloudlet network: A sustainable platform for mobile cloud computing." *IEEE Transactions on Cloud Computing*, **v**ol. 8, Issue: 1, pp.180-192.

[13] A. Boukerche, S. Guan, R.E.De. Grand. (2020). "Sustainable Offloading in Mobile Cloud Computing: Algorithmic Design and Implementation." *ACM Computing Surveys*, vol. 52, Issue 1, no. 11, pp. 1–37, doi.org/10.1145/3286688

[14] E. Ahmed, A. Akhunzada, M. Whaiduzzaman, A. Gani, S.H. Ab Hamid and R. Buyya (2015). Network-centric performance analysis of runtime application migration in mobile cloud computing. *Simul Model Pract Theory*, vol.50, pp.42–56.

[15] G. Shreya, A. Mukherjee, S. Ghosh and R. Buyya. (2019). "Mobi-IoST: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications." *IEEE Transactions on Network Science and Engineering*: DOI 10.1109/TNSE.2019.2941754, IEEE, pp.1-15.

[16] A. Mukherjee, D. Priti, D. De and R. Buyya. (2019). "IoTF2N: "An energy-efficient architectural model for IoT using Femtoletbased fog network." *The Journal of Supercomputing*, vol. 75, no. 11, pp.7125-7146

[17] W. Shi, S. Dustdar. (2016). "The Promise of Edge Computing". *In: The Promise of Edge Computer. Computing*, vol. 49, no. 5, pp. 78–81.

[18] Rida Qayyum, Hina Ejaz, " Data Security in Mobile Cloud Computing: A State of the Art Review", International Journal of Modern Education and Computer Science(IJMECS), Vol.12, No.2, pp. 30-35, 2020.DOI: 10.5815/ijmecs.2020.02.04

[19] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, "Real Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds," *IEEE Trans. Cloud Computing*, vol. 2, no. 2, pp. 168- 180.

[20] G.Yan, S. Wang, A. Zhou, J. Xu, J. Yuan and C. Hsu. (2020). "User allocation-aware edge cloud placement in mobile edge computing." *Software: Practice and Experience*, vol. 50, no. 5, pp. 489-502.

[21] M. Quwaider and Y. Jararweh. (2015). "Cloudlet-based efficient data collection in wireless body area networks". *Simul Model Pract Theory*, vol.50, pp.57–71.

[22] T. Verbelen, P. Simoens, F.D. Turck and B. Dhoedt. (2014). "Adaptive deployment and configuration for mobile augmented reality in the cloudlet". *J Netw Comput Appl*, vol.41, pp. 206–216.

[23] Adil Bashir, Sahil Sholla, " Resource Efficient Security Mechanism for Cloud of Things", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.11, No.4, pp. 41-45, 2021.DOI: 10.5815/ijwmt.2021.04.05

[24] Kapan Oralgazyolu Shakerkhan, Ermek Tolegenovich Abilmazhinov, " Development of a Method for Choosing Cloud Computing on the Platform of Paas for Servicing the State Agencies", International Journal of Modern Education and Computer Science(IJMECS), Vol.11, No.9, pp. 14-25, 2019.DOI: 10.5815/ijmecs.2019.09.02
[25] V. Kovtun, I. Izonin, M. Gregus. (2021). "Mathematical models of the information interaction process in 5G-IoT ecosystem: Different functional scenarios." *ICT Express,* pp. 1-6, /doi.org/10.1016/j.icte.2021.11.008.
[26] H. Mora, M. Gimeno, M. T. Signes-Pont and B. Volckaert. (2019). "Multilayer Architecture Model for Mobile Cloud Computing Paradigm". *Complexity*.Volume, Article ID 3951495, 13 pages. https://doi.org/10.1155/2019/3951495.
[27] A. Mukherjee, D. De and R. G. Roy. (2016). "A power and latency aware cloudlet selection strategy for multi-cloudlet environment". *IEEE Transactions on Cloud Computing*, vol.7, pp.141-154. https://doi.org/10.1109/TCC.2016.2586061.
[28] M. Zhao and K. Zhou. (2019). "Selective Offloading by Exploiting ARIMA-BP for Energy Optimization in Mobile Edge Computing Networks". *Algorithms*, vol.12, no.2, pp.1-13.
[29] S. Banerjee, M. Adhikari, S. Kar and U. Biswas, (2015). "Development and Analysis of A New Cloudlet Allocation Strategy for QoS Improvement in Cloud," *Arabian Journal for Science and Engineering*, vol. 40, no. 5, pp. 1409-1425.
[30] R.G.Alekberov and O.R. Alekperov. (2019). "Procedure of effective use of cloudlets in wireless metropolitan area network environment". *International Journal of Computer Networks & Communications* (IJCNC), vol.11, no.1, pp.93–107.
[31] Manju Mam, Leena G, N S Saxena, "Improved K-means Clustering based Distribution Planning on a Geographical Network", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.4, pp.69-75, 2017. DOI: 10.5815/ijisa.2017.04.08
[32] R.K. Alekberov, (2021). "Challenges of Mobile Devices' Resources and in Communication Channels and their Solutions" International Journal of Computer Network and Information Security (IJCNIS). vol.13, no.1, pp. 39-46.
[33] M. Jia, W. Liang, Z. Xu and M. Huang (2016). "Cloudlet Load Balancing in Wireless Metropolitan Area Networks". *IEEE INFOCOM 2016 – The 35th Annual IEEE International Conference on Computer Communications,* pp.1-9.
[34] G.T. Hicham and E.A. Chaker (2016). "Cloud Computing CPU Allocation and Scheduling Algorithms Using CloudSim Simulator". *International Journal of Electrical and Computer Engineering*, vol.6, no.4, pp.1866-1879. https://doi.org/10.11591/ijece.v6i4.10144.
[35] K. Gai, M. Qiu, H. Zhao, L.Tao and Z. Zong. (2016). "Dynamic Energy-aware Cloudlet-based Mobile Cloud Computing Model for Green Computing". *Journal of Network and Computer Applications*, vol. 59, pp. 46-54.
[36] D. G. Roy, D.De, A. Mukherjee and R. Buyya. (2017). "Application-aware cloudlet selection for computation offloading in multi-cloudlet environment".*J Supercomput* 73: pp. 1672–1690.DOI 10.1007/s11227-016-1872-y.
[37] M. Jia, J. Cao, W. Liang. (2017). "Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks." IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 725-737.

Authors' Profiles

Rashid G. Alakbarov graduated from Automation and Computer Engineering faculty of Azerbaijan Polytechnic University named after C. Ildirim. He received his PhD degree in 2006 from Supreme Attestation Commission under the President of the Republic of Azerbaijan. His primary research interests include various areas in cloud computing, data processing, computer networks, virtual computing, particularly in the area of distributed computing. He is head of department at the Institute of Information Technology as of 2002. Since 2010, he has been leading the development of "AzScienceNet" infrastructure. In 2021, he was appointed an executive director of the institute by the decision of the Presidium of Azerbaijan National Academy of Sciences. He is the author of 95 scientific papers, inclouding 5 inventions.

How to cite this paper: Rashid G. Alakberov, "Clustering Method of Mobile Cloud Computing According to Technical Characteristics of Cloudlets", International Journal of Computer Network and Information Security(IJCNIS), Vol.14, No.3, pp.75-87, 2022. DOI: 10.5815/ijcnis.2022.03.06