FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 994-998 | e-ISSN: 2347-470X

Task Scheduling on Cloudlet in Mobile Cloud Computing with Load Balancing

Poonam[1*] and Suman Sangwan[2]
[1,2]Deenbandhu Chhotu Ram University of Science and Technology, Haryana, India, [2]suman.cse@dcrustm.org.

***Correspondence:** Poonam Ahlawat; ahlawatp12@gmail.com

ABSTRACT- The recent growth in the use of mobile devices has contributed to increased computing and storage requirements. Cloud computing has been used over the past decade to cater to computational and storage needs over the internet. However, the use of various mobile applications like Augmented Reality (AR), M2M Communications, V2X Communications, and the Internet of Things (IoT) led to the emergence of mobile cloud computing (MCC). All data from mobile devices is offloaded and computed on the cloud, removing all limitations incorporated with mobile devices. However, delays induced by the location of data centers led to the birth of edge computing technologies. In this paper, we discuss one of the edge computing technologies, i.e., cloudlet. Cloudlet brings the cloud close to the end-user leading to reduced delay and response time. An algorithm is proposed for scheduling tasks on cloudlet by considering VM's load. Simulation results indicate that the proposed algorithm provides 12% and 29% improvement over EMACS and QRR while balancing the load.

Keywords: Cloud computing, Cloudlet, Edge Computing, Mobile Cloud computing.

ARTICLE INFORMATION
Author(s): Poonam and Suman Sangwan;
Received: 11/07/2022; **Accepted**: 03/11/2022; **Published**: 15/11/2022;
e-ISSN: 2347-470X;
Paper Id: IJEER 1107-05;
Citation: 10.37191/IJEER.100440
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100440.html

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

According to World Advertising Research Center (WARS), around 2 billion people currently access the internet through their mobile phones, which is expected to reach 3.7 billion by 2025. Nowadays, mobile phones are used for all kinds of applications like Augmented Reality, file editing, chatting, video streaming, and gaming. Mainly mobile phones are used for interactive applications [1]. Although recent advances in technologies, mobile phones still have insufficient resources due to restrictions on size, weight, battery, and memory, which requires a technology that can meet all mobile users' demands without compromising resources, processing speed, and delay. Using the cloud for executing mobile applications gives new directions, leading to the concept of mobile cloud computing. Applications of mobile devices can be offloaded on cloud servers for computations and processing at a lower cost [2]. Mobile cloud computing can be defined as an environment where all processing, storage, and execution of mobile applications are done outside the mobile devices, somewhere on the external cloud [3].

Balancing the load of the mobile cloud is also a significant research area. Load balancing is a way of distributing offloaded tasks over all nodes of the cloud uniformly to improve the overall performance of the cloud [4, 5].

Mobile cloud computing is one of the top research areas as it enhances mobile devices' processing capabilities by integrating them with the cloud. The high points of this paper are as follows:

1. Study of existing scheduling and load balancing techniques in MCC.
2. Propose a novel dynamic scheduling technique with load balancing for task scheduling on MCC.
3. The mathematical formulation of load, execution time, and cost.
4. Comparison of the proposed technique with QRR and EMACS.

This study has been structured as follows. *Section 2* describes related work, and *Section 3* discusses cloudlet. *Section 4 and 5* introduce the mathematical model and proposed work, respectively, and *Section 6* discusses the simulation results. Lastly, *Section 7* concludes this paper with future directions.

2. RELATED WORK

Cloudlet is one of the mobile cloud computing architectures whose concept was firstly given by Mahadev Satyanarayanan [2]. It is a middle-tier in three-tier mobile computing architecture and brings cloud resources close to mobile user [5]. Wei et al. [6] proposed an algorithm that minimizes energy consumption while maximizing profit. This algorithm provides approximately 60% better load variation than the random selection scheme in the case of light load. Lin et al. [7] proposed a task scheduling algorithm in a mobile cloud environment that offloads tasks on local cores of mobiles and the cloud. This algorithm minimizes delay with significant energy reduction, and tasks are completed within deadline constraints.

L. Shakkeera [8] proposed energy-aware application scheduling and consolidation algorithm. A hybrid cloud model is proposed to utilize idle resources of nearby mobile devices. This

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 994-998 | e-ISSN: 2347-470X

algorithm provides a significant decrease in energy consumption by application consolidation.

Offloading methods have been addressed on several offloading architectures, such as Honeybee and COSMOS [9]. Based on these architectures, offloading methods are used to enhance the mobile device's performance and save energy. Due to the trade-off of the parameters in the offloading process, the QoS-related offloading methods include network bandwidth, deadline, and power consumption [10]. Various offloading methods like Artificial intelligence-based applications, Directed Acyclic Graphs (DAG) scheduling, Game theory, Lyapunov optimization, Markov Decision Process, and deep learning methods [11, 12] have been applied in various areas. X. Wei et al. [6] proposed HACAS, which balances the system's load in both cases when the load is high, and the load is low, with maximum profit and minimum energy consumption.

3. CLOUDLET

As mobile devices are poor in resources, their execution and storage need to be done outside, like in the cloud. Cloud provides a resource-rich infrastructure on an on-demand basis, eliminating the resource poverty of mobile devices [4]. The main limitation in the use of the cloud by mobile users is long WAN latencies due to multi-hop distance. Cloudlets can solve this problem without being WAN limited. Cloudlet is one of the edge computing technologies [5]. The concept of cloudlet was first introduced by Mahadev Satyanarayanan [2]. Cloudlet is considered a proxy of the central cloud, located somewhere in the middle of the cloud, and mobile users, as in *figure 1* [13].

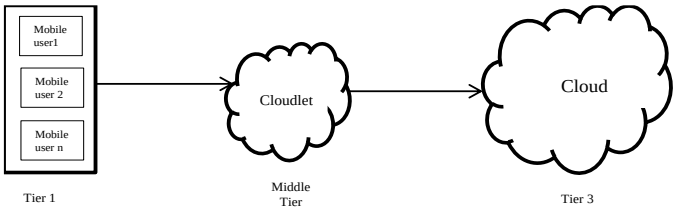

Figure 1: Three-tier architecture

The main objective of cloudlet is to bring the cloud close to mobile users [14]. Cloudlets have decentralized architecture and are dispersed widely. They can be used to cater to the need of nearby mobile devices such as coffee shops and hospitals. Cloudlets need no fixed infrastructure; they can be formed by using nearby resources like nearby mobile phones, and laptops, providing a dynamic infrastructure [15].

4. MATHEMATICAL PROBLEM FORMULATION

For the mathematical formulation of the problem, let us consider *n* tasks arriving for execution on *p* cloudlets and m VMs. Parameters used in this research work are shown in *table 1*.

Table 1: Parameters definition

Parameter	Definition
T_i	i^{th} task
VM_j	j^{th} VM
tl_i	Length of i^{th} task
BW	Bandwidth utilization
CPU	CPU utilization
RAM	RAM utilization
L	System load
Ps	Processing speed
α	No. of processing units
ET_Task	Execution time of task
ET_{VM}	Execution time of VM
TF_i	Finishing time of the i^{th} task
M	Makespan
TC	Total cost

4.1 Load

System load is an indicator of the utilization of system resources, and it is defined in terms of network bandwidth, RAM, and CPU processing [16]. Load is evaluated as shown in *eq. (1).*

$$Load\_VM, L = \frac{1}{1-BW} * \frac{1}{1-RAM} * \frac{1}{1-CPU} \quad (1)$$

4.2 Execution Time of Task

The execution time is the time to execute a task on a VM. It is calculated by using the following *eq. (2).*

$$ET\_Task = \frac{tl}{ps * \alpha} \quad (2)$$

4.3 Execution Time of VM

VM execution time is then given by summing up the minimum execution time of all tasks running on that VM, given by the following *eq. (3).*

$$ET_{VM} = min \sum ET\_Task \quad (3)$$

4.4 Makespan

Makespan is one of the essential criteria that show the highest finishing time among all tasks. Therefore, a low value of makespan means that the task scheduling algorithm is successful in the efficient allocation of tasks to VMs. Generally, makespan is computed as given by *eq. (4)* [17].

$$Makespan, M = max\{TF_i \mid \forall\, i \in List\ of\ Tasks\} \quad (4)$$

4.5 Cost

It is the total cost incurred for the execution of tasks on cloudlet. It is calculated using the following *eq. (5)*

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 994-998 | e-ISSN: 2347-470X

$$Total\_Cost, TC = Data\_Transfer\_Cost + VM\_Cost \quad (5)$$

Where $Data\_Transfer\_Cost$ the cost is incurred in transferring data from mobile to cloud, and VM_Cost is the cost incurred in executing tasks on VM.

4.6 Fitness Function

The fitness function for this scheduling problem can be formed using *eq. (1), (4), and (5)*. Mathematically fitness function is represented using *eq. (6)*.

$$Fitness\ Function, F(x) = \min(a_1 * L + a_2 * M + a_3 * TC) \quad (6)$$

a_1, a_2 and a_3 are constants responsible for optimizing fitness function such that $a_1 + a_2 + a_3 = 1$. Values for these constants are considered as $a_1 = 0.5$, $a_2 = 0.25$ and $a_3 = 0.25$ [18].

5. PROPOSED WORK

A dynamic algorithm has been proposed for scheduling tasks on VMs along with balancing the load. The proposed algorithm consists of two modules, Load_Evaluator and VM_Allocation, for scheduling and balancing the load on the cloudlet. Tasks arrive at cloudlet broker randomly. The resource manager evaluates a load and execution time of all VMs on each cloudlet., and based on it selects the cloudlet.

5.1 Load Evaluator

The primary function of the load evaluator module is to calculate the load of VMs. Load is evaluated by considering bandwidth, memory, and CPU utilization. It is used to check whether VM is overloaded or not. This algorithm is executed repeatedly for auto-generation of balanced VM's list, therefore considering system dynamics. The pseudocode for load evaluation is given as follows:

```
Load_Evaluator()
Input: VM's status
Output: List_BVM
List_VM: Active VM's
Load_VM=1/(1-BW)*1/(1-RAM)*1/(1-CPU)
List_OVM: List of overloaded VMs
List_BVM: List of balanced VMs

Step 1: For i=1 to m in List_VM do
            Find Load_VM[i];
            If Load_VM[i]> Threshold then
                add VM[i] to List_OVM;
            Else
                add VM[i] to List_BVM;
            End if
        End for

Step 2: Sort List_BVM in ascending order of Load_VM
        Go to step 1.
```

5.2 Task Allocation

The Load Evaluator module returns a list of VMs to which tasks can be allocated. Tasks are sorted in increasing order of their task length and allocated to the best fit VM based on VM's load status and execution time. In case all the VMs are overloaded, a new VM is created, and arriving requests are allocated. The pseudocode for task allocation is given as under:

```
Task_Allocation()
Input: List_BVM, List_Task
Output: Allocation of Tasks on VMs
List_Task: List of n Tasks

Step 1: If List_BVM≠0 then
            For i=1 to m in List_BVM do
                Calculate Execution Time of VM[i];
                ET_VM[i]= Execution Time of VM[i];
            End for
            List_Sort= sort ET_VM in ascending order;
            Sort List_Task in increasing order of task length;
            For all Ti∈List_Task and VM∈List_Sort do
      Allocate Tasks to VMs in First come first serve order;
            End for

Step 2: Else
            Create a new VM;
            Allocate Task to new VM;
        End if
```

The proposed algorithm pseudocode is given as follows:

```
Dynamic_Scheduling()
Input: List_Task, List_VM
Output: Tasks Allocation, Balanced VMs
List_Task: List of n Tasks
List_VM: List of m VMs

Step 1: For each task in List_Task do
            For each VM in List_VM do
                Call Load_Evaluator();
                Call Task_Allocation();
            End for
            End for
```

Both Load_Evaluator and Task_Allocation modules are called in this algorithm to carry out the task scheduling while uniformly balancing the load on VMs.

6. RESULTS AND DISCUSSION

In this section, the simulation setup, dataset, and results are discussed.

6.1 Simulation Setup

Simulation has been carried out on a 64-bit Windows 8 machine having Intel Core i3 and 4 GB RAM using the Cloud Analyst tool with Eclipse Java Neon3 IDE. Parameters that are considered for simulation are shown in *table 2*. Tasks are scheduled based on the proposed algorithm, and results are compared with QRR and EMACS algorithms [19]. Simulation is carried out on the CLARKNET dataset [20].

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 994-998 | e-ISSN: 2347-470X

Table 2: Simulation Parameters

Parameters	*Values*
No. of Hosts	3
No. of Cloudlets	10-15
No. of servers	1-5
No. of VMs	5-25
No. of Users	10-80
No. of tasks	50-250
Length of Tasks	100 MI to 500 MI
Storage	1 TB
RAM	2 GB
Processing Speed	50-300 MIPS
Bandwidth	100-200 Mbps

6.2 Makespan

Makespan is the highest finishing time among all tasks. The makespan of all three algorithms is shown in *figure 2*. The proposed algorithm shows an improved makespan compared to others as the number of tasks increases. First, the proposed algorithm allocates tasks to the cloudlet, which are only one hop distance to the user, which leads to reduced makespan in contrast to when tasks are allocated to VMs on cloud servers. Second, Load is the primary parameter in the proposed algorithm. Proposed algorithm assigns tasks to the VM with a smaller value of load and execution time, leading to reduced makespan as compared to QRR and EMACS. All of this means that resource utilization of the proposed algorithm is more uniform than QRR and EMACS algorithms.

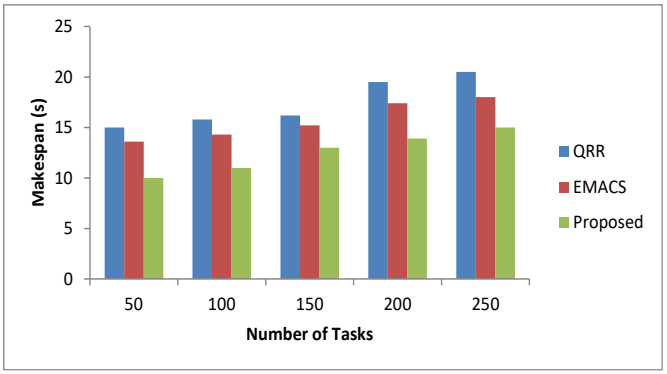

Figure 2: Average Makespan

6.3 Load

System load represents the resource utilization of system resources. The load status of all three VMs for the proposed, QRR and EMACS algorithms is illustrated in *figure 3*. The proposed algorithm achieves better load balancing among VMs than the other two. Tasks are assigned to the VMs by checking their load status to avoid overloaded and underloaded states, thus leading to uniform load distribution among all VMs. EMACS is the second-best algorithm to balance the load among VMs

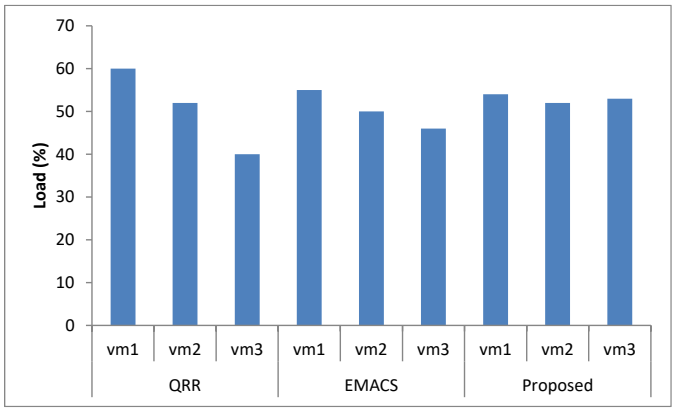

Figure 3: Load Comparison on Three VMs

6.4 Cost

Figure 4 illustrates the comparison of the cost of QRR, EMACS and proposed algorithms. The cost of task includes task transfer cost from a mobile device to the cloudlet and the task execution cost on a VM. In case of cloudlets, transfer time is less than when tasks are offloaded to the cloud. Moreover, tasks are allocated to a VM as per load status leading to uniform load distribution, reduced execution time and costs.

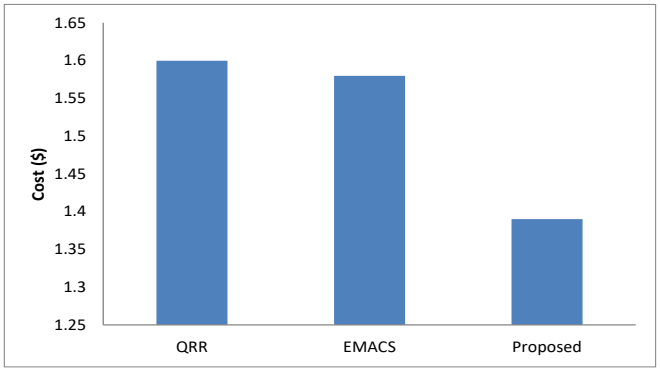

Figure 4: Comparison of Cost

6.5 Drop Rate

Figure 5 represents the drop rate of QRR, EMACS, and the proposed algorithms.

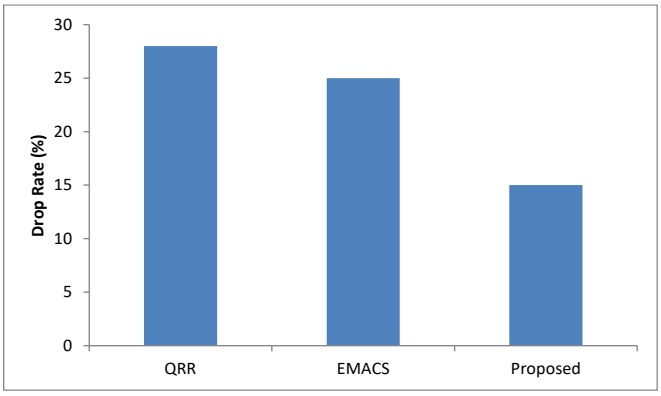

Figure 5: Drop Rate

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 994-998 | e-ISSN: 2347-470X

It is visible that the drop rate of the proposed algorithm is much lower than QRR and EMACS as the proposed algorithm allocates tasks based on the remaining capacity of VMs, which leads to a reduced drop rate.

The simulation results show that the proposed algorithm excels EMACS at about 12% and QRR at about 29%, respectively.

7. CONCLUSION AND FUTURE WORK

A novel dynamic task scheduling technique with load balancing is proposed in the cloudlet environment. This technique schedule tasks on the appropriate VM by considering their load based on memory, bandwidth, and CPU utilization, which are significant resources. Simulation results indicate that under all possible situations proposed technique gives reduced makespan and cost while balancing load than the QRR and EMACS algorithms. The proposed technique provides 12% and 29% improvement over EMACS and QRR algorithms. This work can further be extended as follows: Firstly, by offloading tasks on the cloud when the number of tasks is enormous to be executed on cloudlet. Secondly, task priority can be considered, which matters in some applications. Third, it can be further optimized by applying any optimization technique.

REFERENCES

[1] Dinh, H. T., Lee, C., Niyato, D., Wang, P.:A survey of mobile cloud computing: architecture, applications, and approaches. In: Wireless communications and mobile computing, vol. 13, no. 18, pp. 1587-1611 (2013).

[2] Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. In: IEEE Pervasive Computing, vol. 8, no. 4, pp. 14-23 (2009).

[3] Somula, R., Anilkumar, C., Venkatesh, B., Karrothu, A., Kumar, C. P., Sasikala, R.: Cloudlet services for healthcare applications in mobile cloud computing. In: Proceedings of the 2nd international conference on data engineering and communication technology, Springer, Singapore, pp. 535-543 (2019).

[4] Singh, S.: Load balancing algorithms in cloud computing environment. In: International Journal of Advanced Research in Computer Science, vol. 9, no. 2 (2018).

[5] Nayyer, M. Z., Raza, I., Hussain, S. A.: A survey of cloudlet-based mobile augmentation approaches for resource optimization. In: ACM Computing Surveys (CSUR), vol. 51, no. 5, pp. 1-28 (2018).

[6] Wei, X., Fan, J., Lu, Z., Ding, K.: Application scheduling in mobile cloud computing with load balancing. In: Journal of Applied Mathematics, (2013).

[7] Lin, X., Wang, Y., Xie, Q., Pedram, M.: Energy and performance-aware task scheduling in a mobile cloud computing environment. In: 2014 IEEE 7th international conference on cloud computing, pp. 192-199 (2014).

[8] Shakkeera, L., Tamilselvan, L.: Energy-Aware Application Scheduling and Consolidation in Mobile Cloud Computing with Load Balancing. In: Emerging Research in Computing, Information, Communication and Applications, Springer, New Delhi, pp. 253-264 (2016).

[9] Sangwan, S.: A comparative study of various load balancing algorithms in cloud computing environment. In: IJARET, vol. 11, no. 12, pp. 2735-2760 (2020).

[10] Liao, Z., Ma, Y., Huang, J., Wang, J. Wang, J.: HOTSPOT: A UAV-assisted dynamic mobility-aware offloading for mobile-edge computing in 3-D space. In: IEEE Internet of Things Journal, vol. 8, no. 13, pp. 10940-10952 (2021).

[11] Haris, M., Zubair, S.: Mantaray modified multi-objective Harris hawk optimization algorithm expedites optimal load balancing in cloud computing. In: Journal of King Saud University-Computer and Information Sciences (2021).

[12] Lu, J., Hao, Y., Wu, K., Chen, Y. Wang, Q.: Dynamic offloading for energy-aware scheduling in a mobile cloud. In: Journal of King Saud University-Computer and Information Sciences (2022).

[13] Sangwan, S.: Fuzzy firefly based intelligent algorithm for load balancing in mobile cloud computing. In: Computers, Materials & Continua, vol. 74, no.1, pp. 1783–1799 (2023).

[14] Satyanarayanan, M., Lewis, G., Morris, E., Simanta, S., Boleng, J., Ha, K.: The role of cloudlets in hostile environments. In: IEEE Pervasive Computing, vol. 12, no. 4, pp. 40-49 (2013).

[15] Verbelen, T., Simoens, P., De Turck, F., Dhoedt, B.: Cloudlets: Bringing the cloud to the mobile user. In: Proceedings of the third ACM workshop on Mobile cloud computing and services, pp. 29-36 (2012).

[16] Chen, C., Bao, W., Zhu, X., Ji, H., Xiao, W., Wu, J.: AGILE: A terminal energy efficient scheduling method in mobile cloud computing. In: Transactions on Emerging Telecommunications Technologies, vol. 26, no. 12, pp. 1323-1336 (2015).

[17] Chabbouh, O., Rejeb, S. B., Agoulmine, N., Choukair, Z.: Service scheduling scheme based load balancing for 5G/HetNets Cloud RAN. In: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), pp. 843-849 (2017).

[18] Walia, N. K., Kaur, N., Alowaidi, M., Bhatia, K. S., Mishra, S., Sharma, N. K., Kaur, H.: An energy-efficient hybrid scheduling algorithm for task scheduling in the cloud computing environments. In: IEEE Access, vol. 9, pp. 117325-117337 (2021).

[19] Zhang, C., Yang, Z., He, X., Deng, L.: Multimodal intelligence:Representation learning, information fusion, and applications. In: IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 3, pp. 478-493 (2020).

[20] Rashidi, S., Sharifian, S.: A hybrid heuristic queue based algorithm for task assignment in mobile cloud. In: Future Generation Computer Systems, vol. 68, pp. 331-345 (2017).