International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

Optimization-based Offloading Method for Mobile Cloud Computing Environment

Lakshna Arun, T. N. Ravi

Abstract*: Mobile cloud computing (MCC) is a program that should be applied to defeat the hurdles of computing in the mobile environment. Though developing data-intensive purposes, such as Natural Language Processing (NLP) and face recognition, takes more difficulties on mobile cloud computing stages because of data location and high bandwidth cost issues. To overcome these issues, this paper proposes a dynamic task (resources) allocation model to schedule data-intensive applications on mixed resources (public cloud, cloudlets, and mobile devices) computing environments. Efficient task allocation strategy requires to develop by estimating the number of intentions while performing the decisions of allocation, such as fast response and reduced consumption of energy, to obtain the most reliable task allocation providing the requirements of cloud users and increasing the MCC environment performance. In this paper, Cultural Algorithm (CA) based offloading strategy is proposed for obtaining the minimized task execution time by causing smart decisions for allocation. This proposed algorithm has been implemented using a cloudsim toolkit, and the performance is estimated by analyzing with Genetic and greedy algorithm allocation techniques on a collection of parameters like throughput and makespan for scheduling the resource.

Keywords*: Mobile Cloud Computing, Cloudlets, Resource Allocation, Genetic Algorithm, Cultural Algorithm, Greedy Method.

I. INTRODUCTION

The utilization of mobile devices such as tablets and smartphones experienced a gigantic increment because of the headway in functionalities bolstered by improved highlights, for example, high network, quicker CPU, enormous memory, and advanced sensors. In 2019, it is assessed that mobile gadgets will produce 46% of the web traffic with an expected 30.6 Exabyte created month to month [1]. Albeit mobile devices are presently outfitted with extensive high-performance computation assets, despite everything, they face real difficulties in gathering the prerequisites of data-intensive and mobile applications application computation mobile applications.

Cloud Computing is widely proposed to beat the weaknesses of mobile computing. Cloud computing gives storage as services and computation through an extremely secure and scalable manner.

Revised Manuscript Received on November 30, 2019.
***** Correspondence Author
Lakshna Arun*, Research Scholar, Department of Computer Science, Periyar E.V.R College, Tiruchirappalli, Tamil Nadu, India.
T.N. Ravi, Assistant Professor, Department of Computer Science, Periyar E.V.R College, Tiruchirappalli, Tamil Nadu, India.

The cloud computing model [2] is conceivably the best answer for understanding the incorporation of cloud computing and mobile computing is known as Mobile Cloud Computing(MCC) [3]. MCC intends to increase mobile devices by developing and improving their computing potentials while completing compute-intensive tasks in the resources of the cloud-based environment [4]. Relocating the computations on resource-intensive from mobile accessories to the cloud by the technologies in wireless communication belongs to the theory of offloading in the computation. Computation offloading has been examined in the related works, and many procedures have proposed for meeting consumer Quality of Service (QoS) and optimizing energy consumption through monetary cost and response time. These methods incorporate profiling and context-aware [9], computation of near-by resources [7] and augmentation [5], and providing middleware [8] and cloning of device [6]. While there are a few works about computation offloading, the more significant part of them is restricted to computation-escalated applications which kept on essential MCC condition [10].

II. RELATED WORKS

Nandi, Enakshmi, et al. [11] clarified a proficient mapping of tasks to access resources through a cost-effective way in the MCC environment. The principle point of this paper is to distribute hubs to their separate resources at the cloud server by managing both computation performance and resource cost in MCC. Latiff et al. [12] proposed a Dynamic Clustering League Championship Algorithm (DCLCA) for adaptation to fault tolerance attention to address cloud computing task performance, which would think about the current accessible resources and decrease the less than ideal disappointment of self-sufficient tasks. Tang, Chaogang, et al. [13] displayed the task scheduling issue toward the end-client mobile devices as an optimization problem for energy consumption, while considering information transmission, task reliance, and other limitation conditions, for example, task cost and deadline. The authors further present a few heuristic algorithms to tackle it. Durga, S., et al. [14] proposed a context-aware task scheduling algorithm that expertly apportions the reasonable resources to the customers. A queuing-framework model has given heuristic resources distribution.

Fang, Weiwei, et al. [15] research the resource utilization issue for mobile devices in a multi-client Mobile Edge Computing (MEC) framework with various kinds of conditions of unpredictable channels, arrivals of random and computation tasks.

By together considering the allocation of subcarrier bandwidth and transmit power, scaling of CPU frequency, and scheduling of computation, the authors detail it as a stochastic optimization issue targeting limiting the power utilization of mobiles and to maintain the task-queues long-term stability. By utilizing the optimization method called Lyapunov improvement method, proposed an online control algorithm to illuminate the detailing.

III. PROBLEM IDENTIFICATION

The primary target is to locate the best mobile application offloading plan in which the full energy utilization on the mobile devices and the complete monetary related expense are diminished concerning battery energy and the deadline for an individual task for the available mobile. The offloading plan speaks to tuple for the selected computation environment among neighborhood execution on the mobile devices, and the chose computation condition for the cloud or cloudlet. The optimization issue is figured as a money related cost duplicated by mobile energy utilization because of the presumption of equivalent commitment to the optimization.

IV. PROPOSED CULTURAL ALGORITHM BASED OFFLOADING METHOD

A. Decision-Making Model

In cloud computing conditions, the client applications get performed in a circulated policy. In such a condition, service suppliers would require to ensure the quality of service to the clients, as clients suppose that their task should be provided with cost and least time. Every client application is partitioned into numerous resources, and undertakings should be dispensed for smooth preparing of every one of these tasks, which are disseminated among the Virtual Machines (VM) in the data focus. Each VM may set aside an unusual effort to finish the handling of the task based on the quantity and sort of undertakings it is preparing.

Consequently, before execution, attempts should be allotted and reallocated to required-asset active computing hubs in the cloud condition. In this way, an exact choice must be performed in allocating resources for the compelling approach of tasks based on the approachability of resources to fulfill the client's necessities. Subsequently, the ideal and right primary leadership method for proficient distribution of tasks between the virtual machines is significant for managing execution productivity and more reliable Quality of Service(QoS).

B. The approach in Task Scheduling

Client solicitations arrive in a Poisson distribution pattern and are set into a Queue of Tasks, and these tasks are considered as CLs. The undertaking scheduler distributes resources to these cloudlets utilizing the scheduler pattern to meet the necessities of the clients. The spot-on choice is made to accomplish the best task VM with most extreme preparing effectiveness utilizing genetic procedure, so the undertaking is likewise completed with the base time utilization in the cloud computing framework. Let (VM_1, VM_2, VM_3, VM_4, VM_5… . VM_n) be the arrangement of VMs accessible at the data focus to process the arrangement of tasks or CLs (CL_1, CL_2, CL_3, CL_4, CL_5, CL_6,…, CL_m). Accept that all these VMs are operating in parallel and are connected. These VMs keep running with their resources that are pre-assigned and imparted to different VMs on hosts in the data focus. During the execution, if any tasks encounter the inadequacy of resources, it carries the equivalent to the report of the scheduler, which dynamically reallocates the resources with the VMs and helps in completing the undertaking execution. The quantitative examination is made with the accompanying presumptions:

- The quantitative investigation is inside the scope of code guidance length.
- The landing of tasks is the Poisson distribution.
- Every hub executes one task at a timeslice.

Two scheduling targets are intended to:

- The finishing time of the task is minimized.
- The utilization of the resource is improved.

Computation of Completion Time

In universal cloud conditions, all the computing hubs keep running in parallel that is in Cloudsim condition. All the VMs start at the same time. On the off chance that more than one cloudlet is doled out to one VM, at that point, all the cloudlets will be performed in a steady progression. A two-dimensional (2D) time cluster is accomplished by calculating the Fitness Value (FV) or Fitness Time (FT) of every CL on each VM utilizing fitness function based on the computing strength of VMs. Each VM, the FT, is determined by analyzing the total consummation time of all CLs apportioned to that VM. It is the completion time of the last CL allotted for that VM. The absolute time T devoured by cloudlet j (CL_j) at ith VM $T^i_{CL_j}$ is registered by considering the aggregate of its execution time$ET^i_{CL_j}$, asset reallocation time that is based on the info size $CL_j^{isz}$and output sizeCL_j^{osz} of cloudlet j, and the network Bandwidth B_{wij}. Considering m number of CLs and n number of VMs.

Step by Step Procedure for Minimum Completion Time

Stage 1: *Building a 2D time exhibit of FT or FV of every CL on each VM.*

Stage 2: *The number of cloudlets to be booked for j = 0 to m.*

Stage 3: *The quantity of Virtual Machines For i= 0 to n.*

Stage 4: *Construct the 2D time exhibit with* $FT^i_{CL_j}$

Stage 5: *End for*

Stage 6: *End for*

Stage 7: *ascertaining the completion time of the keep going CL allotted on each VM.*

Stage 8: *number of VMs for I = 0 to n*

Stage 9: *The volume of cloudlets to be booked for j= 0 to m.*

Stage 10: *Completion time of the last CL distributed in the* VM_i.

$$FT_i = \sum_{j=0}^{m} CL_j * E\ (i,j) \quad (1)$$

$$T^i_{CL_j} = ET^i_{CL_j} + \frac{\left(CL_j^{isz} + CL_j^{osz}\right)}{BW_j^i} \quad (2)$$

Stage 11: *End for*

Stage 12: *End for*

Stage 13:*/Allocation of CL to the VM that completes its processing at earlier*
Stage 14: *While (all CLs are alloted to appropriate VM)*
Stage 15: *For each unscheduled cloudlet*
Stage 16: *The quantity of cloudlets to be planned for j = 0 to m.*
Stage 17: *The quantity of virtual machine for i= 0 to n*
Stage 18:*/VM that offers min FT for CL_j. Discover the* $VM_i^{MinFT}(CL_j)$
Stage 19: *when CL_j is doled out to VM_i (i,j) = 1*
Stage 20: *End for*
Stage 21: *End for*

C. Proposed Offloading Approach using Cultural Algorithm

Cultural Algorithm (CA) is utilized to get upgraded clarification from numerous hopeful solutions. The result is taken based on the Fitness Value (FV), which is determined to utilize fitness capacity called a chromosome or individual. A chromosome identifies with one of a kind answer in the clarification space. Further GAs works with an accumulation of chromosomes, known as a population, and utilizations two administrators to be specific mutations and Crossover to create new solutions from existing ones. At that point, reproductions are done to accumulate tasks completing time to keep running on all VMs. Among this time, data CAs give improved answers for the allocation of tasks.

Fitness Function

The fitness capacity is the capacity that surveys the predominance of an individual chromosome, and then the development of the following ages is chosen. Fitness indicates that every individual chromosome and individual with high Fitness has an excellent opportunity to endure. For every age, the fitness value of every person in the population is assessed, higher fitness value people are designated from the present population, at that point, crossover and mutation administrators are utilized to frame another age. The new age of arrangements is then utilized in the following emphasis of the algorithm. The chromosome's overall fitness value is registered to utilize Eq. (3) considering the total time taken to finish the failure probability (β) and schedule (α).

$$Fitness\_Chromosome_i = \alpha\,(Total\_time_i) + \beta(FP_i) \quad (3)$$

where

$Total\_time_i = \sum_{i=j-n} \frac{T\_Len}{VM\_MIPS_i}$, $\alpha + \beta = 1$ VM_MIPS_i is defined as millions of instructions per second for each processor of $VM_j FP_i$, is the network delay between nodes.

Encoding Method

In the encoding strategy, vectors of VMs are considered as chromosomes that is V = [V_1, V_2,… Vi… , V_n] (where n is the quantity of choice factors) to speak to an answer. I is a specific number, goes about as a pointer to the i^{th} VM VM_i to which the j^{th} cloudlet is doled out in the succession of VMs. Each VM in the chromosome signifies the gene with which the CL is running. All out number of conceivable tasks plans accessible are relying upon the quantity of VMs and CLs being utilized. If there are 25 CLs and 12 VMs, at that point, there are 1225 task plans considered to be accessible. The proposed offloading approach chooses the best task plan based on the completing time of the CL to accomplish maximized proficiency.

For the ideal situation considered with 12 VMs and 25 CLs, at first, the 2D time cluster is built by figuring the fitness value (FV) or FT of every candidate (CL) on each VM utilizing fitness capacity based on the computing ability of virtual machines. The proposed offloading approach shapes the chromosome by choosing the fittest VM from each column utilizing the roulette wheel selection strategy. Each line of this exhibit speaks to the length of accessible task plans for every candidate (CL). 12VMs are chosen by the proposed offloading approach as an ideal chromosome for the productive allocation of tasks, as appeared in table 1. V = [11, 8, 4, 5, 4, 0, 2, 1, 7, 11, 5, 2, 3, 5, 4, 4, 6, 11, 10, 3, 2, 4, 4, 5, 6]. That is, the proposed offloading approach chooses the twelfth VM whose succession number of 11 to execute Cloudlet 0 and chooses the ninth VM with arrangement number 8 to execute Cloudlet 1 and so on, the seventh VM is chosen as handling hub of the last CL.

Table 1: Task Allocation in the form of chromosome

CL_0	CL_1	CL_2	CL_3	CL_4	CL_5	⋯	CL_{24}
VM_{11}	VM_8	VM_4	VM_5	VM_4	VM_0	⋯	VM_6

Step by Step Procedure for Proposed Offloading Approach

Input: Crossover point, percentage of mutation, population, fitness function for generation, Gene Length

Output: utilization of the maximum resource, completion time minimization, allocation of the task.

Stage 1: [Start] Determine Random population of n chromosomes, Set POP = pop size; the process starts task portion by picking the best reasonable, proper, free VM.
Stage 2: [Fitness] compute the fitness value of each chromosome in the specified population.
Stage 3: [New population] creates the new population by rehashing the resulting ventures until the formation of the new population is finished.
Step 3.1: [Selection] select two-parent people from the population, compute chromosome with best minimal dispense and fitness value.
Step 3.2: [Crossover] another fittest chromosome value is generated utilizing multi-point Crossover through trading the schedules sets among two chromosomes by utilizing the probability of the new offspring and Crossover by improving the parents is generated. The two fittest chromosomes are chosen.
Step 3.3: [mutation] bits are altered from 0 to 1 or 1 to 0 with a probability of mutation Pm. Transform a visit utilizing swap mutation. With the probability of mutation, change the new child at certain positions.
Step 3.4: [Accept] Place new offspring as new population and utilize this population for the next round of the cycle.
Stage 4: [Exchange] utilize the new generation as the current generation.
Stage 5: [Test] if the leave condition is fulfilled, at that point, part of the bargain return the person to the chromosome with the least fitness value is chosen for the timetable.
Stage 6: [Loop] go to stage 2

V. EXPERIMENTAL RESULT AND DISCUSSION

This segment represents the exploratory arrangement and the outcomes got with the proposed offloading approach. The proposed offloading approach is surveyed in Java Cloudsim. The main record is executed in the default bundle, which acknowledges the contribution as the system bandwidth and delay and amount of errands. The amount of errands suggests the amount of cloudlets that can be changed by changing the incentive to make cloudlets () work. The algorithm is tried by setting the parameters as appeared in Table 2. Simulation has been completed with two distinct cases, thinking about differed sets of CLs and VMs for each case. The choice must be made to recognize which cloudlet can be executed in the VM and to what extent it takes to finish it. Time utilization may fluctuate for various cloudlets running on the equivalent VM, and likewise, the same tasks may set aside extraordinary effort to complete its execution in various VMs. Quicker the VM lesser the allocation execution time. More significant the tasks more the time it devours and with great bandwidth of the network.

Table 2: Simulation Setup

Parameters	Value
Population size	100
Maximum Evaluation	500
Cross over operator	Single Point
Cross over probability	Pc
Mutation Operator	Bitflip
Mutation rate	0.15
Knowledgebase	Normative Knowledge

Table 3: Allocation of CloudLet with time consumption by Genetic Algorithm, Greedy Algorithm and Proposed offloading Approach

Virtual ID	Genetic Algorithm		Greedy Algorithm		Proposed Offloading Approach	
	Cloudlet ID	*Time*	*Cloudlet ID*	*Time*	*Cloudle t ID*	*Time*
0	1	1.79	0	5.46	1	0.46
1	0	3.51	1	5.06	10	2.7
2	2	2.28	2	6.99	2	1.13
3	4	3.11	3	8.86	11	0.37
4	3	7.75	4	8.89	4	0.41
5	4	5.69	5	9.17	9	0.19
6	5	6.67	6	11.9 4	11	0.66
7	7	6.14	7	10.1 1	2	0.42
8	8	8.79	8	10.3 7	6	0.79
9	9	4.4	9	9.35	8	0.35
10	10	3.12	10	9.44	3	0.44
11	10	2.01	11	8.09	3	1.51
12	11	2.87	0	6.57	5	2.76
13	5	2.22	1	5.44	7	1.23
14	3	2.05	2	4.45	0	0.48

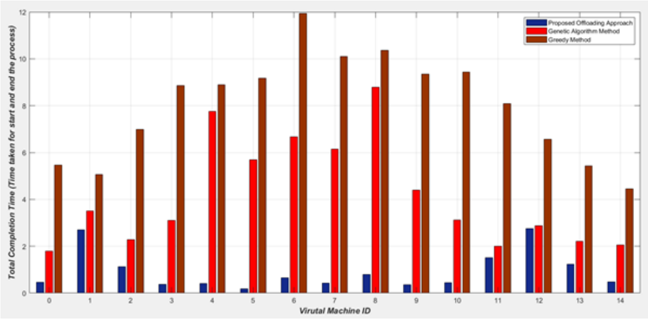

FIGURE 1: Total Completion time of the proposed offloading approach, Genetic Algorithm based offloading approach and Greedy Method for given Virtual Machine ID

VI. CONCLUSION

Exactly when the colossal proportion of information enters the cloud, the exceptional weight of the cloud structure augments as the massive proportion of information ought to be handled and put away. Basic leadership for the distribution of tasks is one of the basic issues in the MCC condition. In this paper, the issue has been tended to by realizing the proposed offloading approach for getting the perfect assignment of resources. The proposed offloading approach has improved the introduction of the versatile cloud computing system using cloudlets where the distribution of resources was finished using irregular and heuristic pursuit by the cloud server. The results got by the proposed technique is conceivable with satisfactory execution stood out from various approaches. Further work could be considered with parameters, for instance, prerequisites of resources and utilization of energy in the basic decision strategy.

REFERENCES

1. Miettinen, Antti P., and Jukka K. Nurminen. "Energy efficiency of mobile clients in cloud computing." *HotCloud* 10.4-4 (2010): 19.
2. Wen, Yonggang, Weiwen Zhang, and Haiyun Luo. "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones." *2012 Proceedings Ieee Infocom*. IEEE, 2012.
3. Park, JiSu, HeonChang Yu, and EunYoung Lee. "Resource allocation techniques based on availability and movement reliability for mobile cloud computing." *International Conference on Distributed Computing and Internet Technology*. Springer, Berlin, Heidelberg, 263-264, 2012.
4. Chen, Min, et al. "Cloud-based wireless network: Virtualized, reconfigurable, smart wireless network to enable 5G technologies." *Mobile Networks and Applications* 20.6 (2015): 704-712.
5. Wan, Jiafu, et al. "VCMIA: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing." *Mobile Networks and Applications* 19.2 (2014): 153-160.
6. Wen, Yonggang, Weiwen Zhang, and Haiyun Luo. "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones." *2012 Proceedings Ieee Infocom*. IEEE, 2716-2720, 2012.
7. Cardellini, Valeria, et al. "A game-theoretic approach to computation offloading in mobile cloud computing." *Mathematical Programming* 157.2 (2016): 421-449.
8. Srirama, Satish Narayana, Eero Vainikko, and Matthias Jarke. "Supporting mobile web service provisioning with cloud computing." (2010).
9. Khan, Zaheer, Saad Liaquat Kiani, and Kamran Soomro. "A framework for cloud-based context-aware information services for citizens in smart cities." *Journal of Cloud Computing* 3.1 (2014): 14.

10. Wu, Huaming, et al. "An optimal offloading partitioning algorithm in mobile cloud computing." *International Conference on Quantitative Evaluation of Systems*. Springer, Cham, 2016.
11. Nandi, Enakshmi, et al. "Improved Cost-Effective Technique for Resource Allocation in Mobile Cloud Computing." *Progress in Computing, Analytics, and Networking*. Springer, Singapore, 2018. 551-558.
12. Latiff, Muhammad Shafie Abd, Syed Hamid Hussain Madni, and Mohammed Abdullahi. "Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm." *Neural Computing and Applications* 29.1 (2018): 279-293.
13. Tang, Chaogang, et al. "Energy-aware task scheduling in mobile cloud computing." *Distributed and Parallel Databases*36.3 (2018): 529-553.
14. Durga, S., S. Mohan, and J. Dinesh Peter. "A Two-Stage Queue Model for Context-Aware Task Scheduling in Mobile Multimedia Cloud Environments." *Advances in Big Data and Cloud Computing*. Springer, Singapore, 2018. 287-297.
15. Fang, Weiwei, et al. "OKRA: optimal task and resource allocation for energy minimization in mobile edge computing systems." *Wireless Networks* 25.5 (2019): 2851-2867.