LATENCY OPTIMIZATION IN COMPUTATION PARTITIONING OF MULTIUSER MOBILE CLOUD APPLICATIONS

M S Patel[1], Pooja R[2], Akshaya B M[3], Natasha Susan George[4]

[1]*Associate Professor, Computer Science Department, TJIT, Karnataka, India*
[2]*Pursuing B.Tech, Computer Science Department, TJIT, Karnataka, India*
[3]*Pursuing B.Tech, Computer Science Department, TJIT, Karnataka, India*
[4]*Pursuing B.Tech, Computer Science Department, TJIT, Karnataka, India*

Abstract

In mobile cloud computing, dividing the computations between the mobile devices and the cloud is very significant and Essential. In the existing work , these divisions of computations exists for a single user only . That is, it focuses on to optimize the time of the applications for that particular user only. Such existing works makes an assumption that there are enough resources always available on the cloud for the user to make the computations of the tasks which are offloaded on to the cloud. But this might not be an ideal approach when we consider several users who are trying to offload their tasks on to the cloud. There may arise situations where the resources on the cloud may not be available for the user's computation to be executed, and hence the user is forced to wait until the resource gets available and only after that the task gets executed. And this results in scheduling latency of the application. The Scheduling latency which when not considered, results in the Performance deprivation. Hence we propose a method in this paper known as Multiple user Partitioning of computations Problem (MPCP). The MPCP approach emphasizes on the divisions of computations of multiple user's and scheduling the computations that are offloaded on to the resources that are present in the cloud. Unlike the existing works which focuses to minimize the completion time of application for single user, our work aims to minimize the average completion time for all the users that take the services of the cloud.

Keywords: *Computations, Scheduling Latency, and Optimize, etc…*

----------***----------

1. INTRODUCTION

New type of mobile applications such as healthcare monitoring, mobile biometric, augmented reality, etc are emerging due to the rapid growth of sensors being used in smart phones. Sensors such as cameras, GPS, accelerometers, etc need high rate processing in the mobile. This processing affects the performance of the mobile . To overcome the issue the concept of scheduling was introduced. In this approach the tasks of the mobile are offloaded onto the cloud. The scheme of scheduling is to execute the tasks on the resources available on the cloud. This enables the tasks to be completed faster.

Partitioning is the crucial issue to be considered before scheduling the tasks. Division of tasks between the mobile device and cloud is what the partitioning is concerned with. Generally any mobile application consists of several tasks which are dependent on each other. A decision is made by the partitioner which of these must be offloaded to cloud and which all to be executed on the mobile device itself. The method effectively results in minimization of cost. Existing works [4],[5],[6],[7],[8],[9],[10],[12] varies by their cost models. Factors such as data transmission amount, energy consumption, application delay of network, etc are considered while creating such models.

Nevertheless the existing works focus the single user's partitioning. Here partitioning is done for that particular user thereby ignoring the other users' partition results. In this approach an assumption is made that the cloud resource is immediately allotted for user as soon as the task is being offloaded to cloud. This might seem appealing when we consider only one user. But viewing is practically, the approach does not seem ideal.

There will be numerous users who offload their tasks onto the cloud. Cloud infrastructure has limited number of resources. Therefore when several users offload their tasks onto the cloud, situation arises when there are no resources available. This forces the users to wait for their tasks to put on to the cloud resources. Hence there is a scheduling delay. To overcome this drawback, we come up with Multiple user Partitioning of computations Problem (MPCP).

In MPCP the scheduling of tasks which are offloaded onto the cloud does not depend only on one single user but depends on all the users since there is a competition between all the users for cloud resources. One of the example that we can take is, decision making of user's task being offloaded does not depend only on device overhead and operational cost but also depends on the load of the cloud. Load is nothing but the number of users that have offloaded their tasks onto the cloud. Whenever there is large value of load, the user who wishes to offload the task may sacrifice to execute the task on mobile device itself instead of wasting the time by waiting for resource to be available affecting performance.

In our paper the MPCP is studied. The two main modules of this approach is Partitioning and scheduling. Initially the partition of tasks of users are done where a decision is made as to which tasks have to be done in mobile devices and which have to be offloaded to resources of cloud. Our main aim is to minimize the average delay of the application. We have considered the application delay since delay is a critical attribute for mobile applications which are latency sensitive.

2.RELATED WORK

We brief out existing related works on partitioning of computations in mobile cloud computing.

The exploitation of cloud services has led to a new field of mobile cloud computing.The approaches of mobile cloud computing can beclassified into three main categories based on the pre vious works[4]- (i) mobile dvices getting extended access to services of cloud. (ii)collaboration of mobile devices to work as providers of cloud resources. (iii) diving the execution of mobile tasks onto the resources of cloud.

Out of the approaches specified, the approach that is majorly popular is the third one [4],[5],[6],[7],[9].The main focus is given to the partition or divisions of computations. However the decision of offloading the task of a device is viewed only locally i.e, it does not consider about the other devices' tasks.[11] focuses on reducing the consumption of energy whereas [12] focuses to exploit throughput and time of execution.

The decisions of offloading the tasks of a device globally i.e taking into consideration of other tasks of other devices'[4],[5],[6],[8], gives better result than the one where it does locally. [5] Exhibits the above approach.In all the above works though they try to optimize, they vary in their models of applications.

In the recent works [4],[5],[6],[7],[8],[9],[11],[12] the divisions of computations are done only considering single user view.Also they assume that there are resources available all the time in the cloud. Some of the existing works throws light upon how when there are large number of users offloading their tasks to clouds, resources become unavailable. Therefore in our paper we propose a concept of multiple user partitioning of computation problems.

3.SYSTEM MODEL AND PROBLEM FORMULATION

3.1 Application Model

Mobile device applications which are sensitive to delay are our main target. In such applications, data is taken as input, operations are performed on data and then the output is given. Let us take the example of mobile virtual reality.

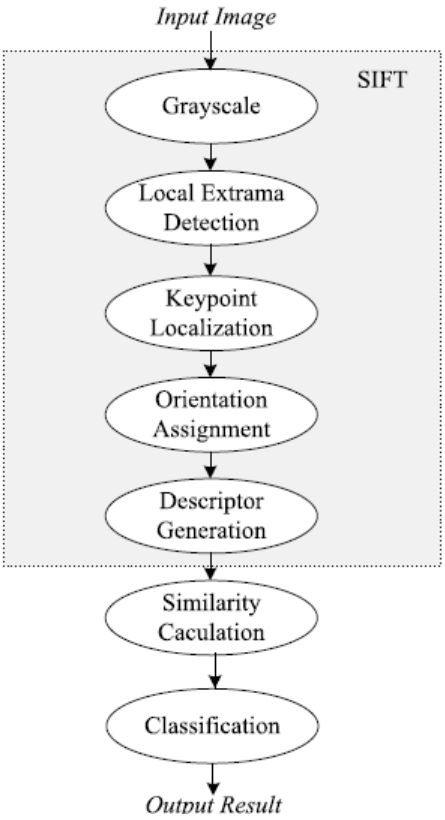

Fig 3.1: The functional module of picture based object realization.

Here, Generally the input is taken through sensors and cameras about the environment then a series of operations are performed continuously based on user's input. The main part of mobile virtual reality is image based. The image based recognition modules are as shown in the fig 3.1.

The mobile applications are divided into several tasks or modules.Every module is dependant on each other. That is, a module's output is given as input to the consequent module.And hence we cannot jump into processing of a module until the previous module has completed its processing. The modules can be executed either on the mobile device or it can be offloaded to the cloud for execution. Once the processing is done, the results are to be given to the mobile application from cloud.For example let us consider a gaming application. The game consists of many levels, here each level is considered as a module. Until previous level completes its execution, the next level can nt be processed or played. Therefore we can say that the modules are interdependant.So a level's computation can be executed in the mobile device or it can be offloaded to the cloud for executing the computation.Once he results are obtainedto the mobile, the next level can be started.It continous recursively. The application's execution time is the performance metric. Since the responsive latency or delay is represented by the execution time, the term latency is used in our paper.The latency is the total of all the modules' computation time and the transmission of data between the modules.

3.2 SCPC(SINGLE USER PARTITIONING OF COMPUTATIONS)

Single user partitioning of computations is the existing mobile cloud computation approach which takes into consideration of the single user only. Here the user requests for cloud service to run his application. Suppose the application consists of n number of modules, the module has two options, (i) The module may be executed in the mobile device itself or (ii) The module can be executed on the cloud resource by offloading it on to the cloud.For module p, let c_p be the execution time on the cloud($1 \le p \le n$) and $w_{p\ be}$ the execution time when it is done on the mobile device.w_p is greater than c_p.

P and p+1 are two adjacent modules where one run in mobile side and one run in cloud side. π_p is the time of data transmission. If p and p+1 run on same side then the value of π_p will be equal to zero. 0 and n+1 are two module which are entry module and exit module to show the direction of data transmission i.e, to/from the mobile.

Definition1 (Single user partitioning of computations problem).
Here the SPCP is used to determine the module that is supposed to be offloaded onto the cloud given that the computation cost is c_p and $w_p(1 \le p \le n)$ and the communication cost $\prod_p(0 \le p \le n)$,the SPCP is used so as to compute which module should be offloaded onto the cloud so that the delay is minimised It is given by,

$$\min d = \sum_{i=1}^{n} [(1 - x_p)w_p + x_p c_p] + \sum_{p=0}^{n} |x_p - x_{p+1}|\pi_p$$

given x_p is termed as a binary constant, $x_p=1$ if the job p is offloaded onto the cloud or else $x_0=0$;and $x_0=x_{n+1}=0$.

4. MULTIPLE USER PARTITIONING OF COMPUTATION PROBLEM (MPCP)

Fig 4.1. shows the model of the system of multiple user partitioning of computation problem. The two important parts of the system are the cloud and the mobile. The application is present on the mobile. The client middleware consists of a monitor agent. The monitor agent collects the information of the mobile device such as speed, RAM memory etc which is forwarded to the partitioner that is present on the PaaS(Platform as a service) of the cloud server.

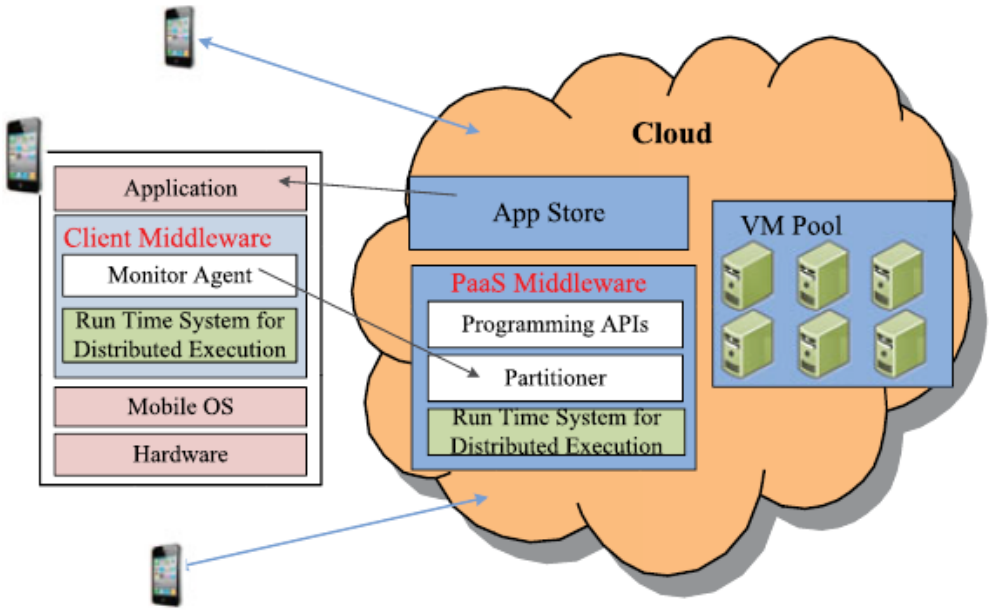

Fig 4.1: System design of MPCP

The virtual machine pool is present on the cloud. Initially when the user registers to the cloud as logs in the application starts. The tasks of the mobile application are taken as modules. In the mobile device, the tasks are scheduled as to whether it has to be executed on mobile or offloaded to the cloud. The monitor agent which monitors these tasks, collects the CPU and RAM information of the user's mobile device and sends it to the partitioner requesting the service of the cloud. The partitioner collects the information of all the tasks from various user's who have requested for cloud service. The partitioner based on the information gathered such as speed and memory of the mobile device along with the resource available on the cloud, decides whether it can be executed in cloud or should it be sent back to the mobile device itself. Once it is done deciding ,it sends instructions to the monitor agent regarding the same. Now the monitor agent must re-adjust itself and reschedule the rejected modules in the mobile device which were sent back from partitioner. The accepted tasks or modules are offloaded into the virtual machine pool.

All the tasks that are present in the virtual machine pool must be scheduled. Scheduling is nothing but deciding which task of which user must be executed. The scheduler schedules the task for execution. Once the execution is done, The results are sent back to the application of the user.

5. FIND AND MODIFY

Since at certain times, the resource are not available for offloading the tasks. The resource constraints appear. As the name suggests, FINDANDMODIFY, finds the time intervals where resource constraints occurs and modifies them. For example, a task t_1 is sent to the cloud for cloud service. The FINDANDMODIFY which is implemented in the cloud checks for all the time intervals. Suppose the resources are unavailable for task t_1 and the resource constraint is said to be violated. Now, the FINDANDMODIFY will modify the task t_1 saying that it has to be executed on the mobile device itself and cannot be offloaded to the cloud. The modifying of the scheduled is done in a greedy way which can release long occupation period of resources and thereby decreasing the average delay of the applications of multiple users.

6. CONCLUSIONS

To overcome the drawbacks of the existing system of the mobile cloud computing which focuses only of single-user partitioning of computations, we propose an approach in this paper for multiple user partitioning of computations. This method's two main tasks are partitioning and scheduling. The tasks of multiple user's are partitioned again in the cloud server by the partitioner and scheduled for execution. The FINDANDMODIFY methods suggested helps in repartitioning of the tasks/modules which tries to minimize the average delay of application of multiple users.

REFERENCES

[1]. LeiYang,JiannongCao,Senior Member, IEEE,
Hui Cheng, Member, IEEE, and Yusheng Ji, Member, IEEE
Multi-User Computation Partitioning for Latency
Sensitive Mobile Cloud Applications.
[2]. E. Cuervoy, A. Balasubramanianz, and D. Cho, "MAUI: Making smartphones last longer with code offload," in Proc. 8th Int. Conf.Mobile Syst., Appl. Services, 2010, pp. 49–62.
[3]. X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," Mobile Netw. Appl.,vol. 16, no. 3, pp. 379–394, 2009.
[4]. U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "A cost-aware elasticity provisioning system for the cloud," in Proc. 31st Int. Conf. Distrib. Comput. Syst., 2011, pp. 559–570.
[5].G. Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in Proc. 1st ACM Workshop Mobile Cloud Computer.Services: Social Netw. Beyond, 2010, pp. 37–41.
[6]. E. E. Marinelli, "Hyrax: Cloud computing on mobile devices using MapReduce," Master thesis, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2009.
[7]. L. Yang, J. Cao, S. Tang, T. Li, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in Proc. IEEE 5th Int. Conf. Cloud Comput., 2012, pp. 794–802.
[8]. K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy," IEEE Comput., vol. 43, no. 4, pp. 51–56, Apr. 2010.
[9]. Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: A partition scheme," in Proc. Int.Conf. Compilers, Archit. Synthesis Embedded Syst., 2001, pp. 238–246.
[10]. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.
[11]. U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "A cost-aware elasticity provisioning system for the cloud," in Proc. 31st Int. Conf. Distrib. Comput. Syst., 2011, pp. 559–570.
[12] G. Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in Proc. 1st ACM Workshop Mobile Cloud Comput. Services: Social Netw. Beyond, 2010, pp. 37–41.