Task Schedule Problem for Resource Augmentation in Mobile Cloud Computing

[1]M. Maheswari
[1]Student,
Department of Computer Science and Engineering,
M.A.R College of Engineering and Technology,
Trichy, Tamil Nadu, India

[2] K. Sivanandham
[2] Assistant Professor,
Department of Computer Science and Engineering,
M.A.R College of Engineering and Technology,
Trichyi, Tamil Nadu, India

***Abstract-* We proposed to utilize a centralized broker-node to perform task scheduling for the resource augmentation of a large number of mobile devices. The task scheduler model focused on energy optimization was proposed for the centralized task scheduling problem. In this paper, the model extends the optimization process by including an economic element to it. Thus, we propose an energy and monetary cost-aware mathematical task scheduler model. Compared to the previous model, this model, can allow mobile devices to offload multiple tasks to cloud resources. The results in this paper are more thorough and more aspects of task offloading have been analysed. For instance, the model is evaluated under two different resource augmentation environments for mobile cloud computing: a local private cloud and public clouds. More precisely, the task scheduling problem is optimally solved to minimize: (i) the total energy consumption when applied to a local private cloud, and (ii) the total energy consumption and monetary cost when applied to public clouds. Our proposed model at the centralized broker-node finds optimal solutions for task assignment problem, and provides a Significant reduction in the total costs compared with the task assignment by the centralized scheduler without optimization.**

Keywords- Mobile cloud computing, Cloud computing, Task scheduler, computation offloading, resource augmentation.

I. INTRODUCTION

Advancements in computing hardware and communication technologies have enabled mobile devices to support resource intensive applications. However, resource constraints intrinsic to their size and weight put limits and most of the time the desired performance cannot be achieved, and/or their battery will drain faster compared to normal usage. Among the various resource augmentation approaches, task offloading is an approach that enables mobile devices to support resource intensive applications by remotely executing them on rich computing nodes. In a Resource Augmentation Environment (RAE), mobile devices can decide to offload tasks when their available resources are not adequate either to execute their tasks or to achieve the desired performance (i.e. small execution time and/or low use of battery energy).

The resource augmentation of mobile devices through task offloading poses some challenges. Task offloading involves additional data communication, which may increase the task's remote completion time and/or energy consumption, and may incur a monetary cost for using resources at a remote location. Thus, to determine whether task offloading is beneficial or not, a task scheduling process checks if the cost of executing it locally on the mobile device. Therefore, prior to offloading, monitoring process would repeatedly contact the remote computation nodes to obtain an up-to-date status of currently available resources such that an appropriate offloading decision can be made. However, employing such a scheme has limitations, in a scenario, in which a large number of mobile devices and multiple remote computing nodes.

In our recent works, the proposed centralized broker node in was utilized to manage task scheduling on behalf of a large number of mobile devices. In these works, a mathematical model, subject to various constraints was proposed for the centralized task scheduling problem. The model was evaluated in a large Cyber Foraging System (CFS) [1] and in Mobile Cloud Computing (MCC) environments [2]. The model provides optimal solutions for the task scheduling problem by minimizing the total energy consumption across all mobile devices in the system.

The main contributions in this paper are as follows. This paper focuses on mobile cloud computing, which an emerging research topic in the field of cloud is computing. In a mobile cloud computing environment, we utilize centralized broker node architecture proposed in our previous work [9] for the resource augmentation of a large number of mobile devices The model in our previous works [1], [2] was focused on energy optimization. Indeed, offloading decisions can now be made based on energy consumption, monetary cost, or a combination of the two. Thus, a mathematical model for the centralized task scheduling problem is proposed, which
Minimizes: (i) the total energy consumption when applied to a
local private cloud, and (ii) the total energy consumption and the total monetary cost when applied to public clouds. We consider monetary cost of using computing and network resources not the initial purchase cost. Therefore, the monetary cost when using resources from a private cloud is assumed to be zero. In the model, user-defined delay tolerance is considered for every task, which puts a limit on the delay of the offloading tasks. In turn, the delay of a task defines the constraint for the minimum required data rate for a given input/output data sizes of the task.

II. PROBLEM DEFINITIONS

A resource augmentation environment for a large number of mobile devices, (ii) utilize a centralized broker-node to centrally handle task scheduling on behalf of a large number of mobile devices, and (iii) propose an energy and monetary cost-aware mathematical model for the centralized task scheduling problem. Existing works related to resource augmentation through task offloading have considered offloading from a single mobile device onto a server.

However, found that the task offloading requirements by a large number of mobile devices pose challenges. Therefore, in this work, in which a large number of mobile devices and multiple service nodes are expected. The results in suggests that the resource monitoring time is smaller and the scalability of the system is better when resource monitoring is performed by a centralized broker-node compared with when it is performed individually by a large number of mobile devices.

Therefore, in this paper, a centralized broker-node is utilized to perform task scheduling on behalf of a large number of mobile devices. A centralized node is utilized for a solver, which provides optimal solutions for a task partitioning problem by minimizing the memory usage of the device. Which helps mobile devices in finding a particular service among the multiple service nodes in the system. There are cloud computing environments, such as those in that utilize a broker-node to find service providers and negotiate for the required resources on behalf of mobile devices. Unlike task scheduler models for hybrid cloud computing environments, our proposed model does not facilitate dynamic switching between a local cloud and public clouds.

III. OUR CONTRIBUTION

In the first environment, computation resources are available from a local private cloud, which is accessible to mobile devices through a WiFi network, as shown in Figure 1(a). On the other hand, computation resources in the second environment are available from public clouds, which are accessible through the Internet, as shown in Figure 1(b).

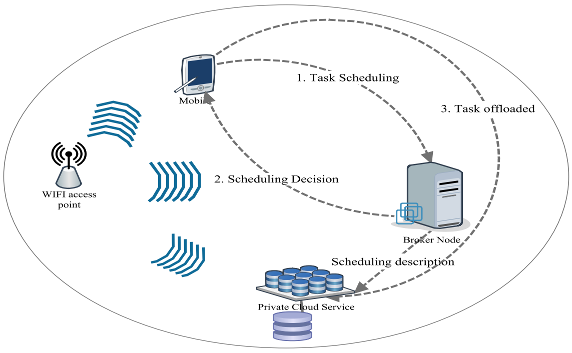

Fig.1(a) RAE using a local private cloud

Paying for resources: Mobile users do not pay when using computation resources from the local private cloud; however, they pay when using resources from the public clouds. Thus, offloading a task onto the local private cloud involves only the energy consumption when transferring data. On the other hand, offloading onto the public clouds involves: (i) energy consumption incurred when transferring data, and (ii) monetary cost incurred when using computing resources per unit time, and transferring data per unit bytes. Availability of resources: The computation resources available from the local private cloud are limited. Therefore, when a large number of mobile devices requests task offloading, at some point some tasks will be denied offloading due to the finite amount of resources. On the other hand, infinite computation resources are available from public clouds. Thus, when offloading tasks from any number of mobile devices, none of the tasks will be denied offloading due to a lack of resources.

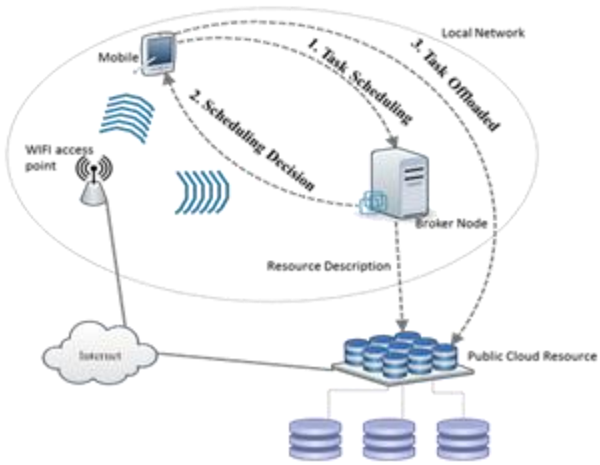

Fig.1(b) RAE using a local public cloud

Accessibility of resources: The computation resources from the local private cloud are present in the vicinity of the mobile devices and are accessed through a WiFi network. However, the resources from public clouds are at WAN latency from the mobile devices and are accessed through the Internet. Therefore, the mobile devices experience a higher data rate when accessing resources from the local private cloud than from the public clouds

This paper is organized as follows: Section 4 gives an overview of the system. Section 5 provides experimental results. Section 6 covers related works, while Section 7 concludes the paper.

IV. OVERVIEW OF OUR APPROACH

The proposed task scheduler model is evaluated under two different resource augmentation environments for mobile cloud computing, as shown in Figure 1. When applied to RAE using a local private cloud (Figure 1(a)), the model finds an optimal solution for the total energy consumption across all mobile devices in the system. When applied to RAE using public clouds (Figure 1(b)), the model finds an optimal solution for the total energy consumption and the total monetary cost across all mobile devices in the system. To this end, a mathematical model based on the following assumptions and notation is proposed.

A. Task Scheduling

When offloading a task, the mobile device may have various offloading goals set by the user, which may include: (i) saving energy on the mobile device, (ii) saving monetary cost of using computation resources, (iii) enhancing the task execution time, and (iv) achieving any combination of the above. A task scheduler decides whether offloading is beneficial or not based on the goals and current status of the availability and requirement of resources on the mobile device and the remote computation nodes. Based on the decision, the task is offloaded either onto a remote computation node, or it is executed locally on the mobile device. However, a resource augmentation environment is a dynamic environment. The requirements for resources may change for a task with a change in its input data, and/or a change in the offloading goals (delay time, battery consumption, etc.). The availability of the resources may also change at remote computation nodes (as is the case for resources from a private cloud, i.e. available CPU power, memory, etc.), and at the wireless network (bandwidth, network latency, etc.). Therefore, it is imperative to decide on the remote execution location dynamically based on the current requirements and availability of resources. In this case, inside the task scheduler, the resource monitoring process is triggered periodically. As explained earlier, periodic resource monitoring by a large number of mobile devices poses challenges; thus, performing resource monitoring on behalf of all the mobile devices at a centralized node was proposed.

$$\alpha = \frac{wep\ \_\ \text{total energy consumption}}{\text{total monetary cost}}\ ;\ wep\ \_\ 0:$$

B. Using a Centralized Broker

In our previous work, performing task scheduling at a centralized node on behalf of all mobile devices in the system,
was proposed. The centralized node was referred to as broker node. Also, in this work, the location of the task scheduler in both RAEs is the centralized broker-node, as shown in Figure 1. The various steps involved while scheduling tasks from mobile devices onto the resources in a cloud are also illustrated in the figure, which include: (1) mobile devices contact the centralized task scheduling service at the broker-node, (2) the task scheduler decides the appropriate offloading location on behalf of the mobile devices by minimizing the total energy consumption and the total monetary cost across all mobile devices subject to various constraints, and based on the task scheduling decision, each task is either offloaded to resources on the cloud or it is executed locally on the mobile device. Multiple broker nodes, load balancing, and reliability should be part of the system design in order to avoid having a single point of failure. However, these issues of load balancing and reliability along with security and access authentication to cloud providers are outside the scope of this paper.

$$d_{kmjvcir} = t^{send}_{kmjvcir} + t^{exec}_{kmjvcir} + t^{rec}_{kmjvcir}\ (1)$$

Notation the following notation is composed of sets, cost parameters, constraints parameters, and decision variables of the model. Sets M, the set of all mobile devices, where mobile device m 2M.Km, the set of all tasks in mobile device m 2 M, where task kmj 2 Km and j : 1 ! jKmj.C, the set of cloud providers, where cloud provider c 2 C.c, number of CPUs available at cloud provider c 2 C.c, amount of memory available at cloud provider c 2 C.I, the set of all VM instance types in each cloud provider in C, where VM instance type i 2 I.Vci, the set of all VMs of instance type i 2 I from cloud provider c 2 C, where VM vcir 2 Vci and r : 1 ! jVcij.Fvcir , speed-up factor of VM vcir with respect to the execution speed of each of the mobile device in M.bvcir , available data rate between VM vcir and each of the mobile device in M.
Cost Parameters
_ etkmj , energy consumed (in joules) when task kmj is executed locally on mobile device m.
_ etkmjvcir , energy consumed (in joules) when task kmj is executed remotely on VM vcir.
_ skmj , required number of slots of unit time for task kmj when executed remotely.
_ pin c , monetary cost of unit bytes of inbound traffic to cloud provider c 2 C.
_ pout c , monetary cost of unit bytes of outbound traffic from
cloud provider c 2 C.
_ pres vcir , monetary cost per unit time period of using VM vcir of instance type i 2 I from cloud provider c 2 C.
Constraints Parameters
_ texec kmj , execution time of task kmj when executed locally on mobile device m 2M.
_ texec kmjvcir , execution time of task kmj when executed remotely on VM vcir.
kmj , amount of input data required to process task kmj.
kmj , amount of output data generated by the execution of task kmj.
_ d kmjvcir , delay of task kmj when executed on VM vcir. The task's delay includes the following components: tsend kmjvcir , time to send input data (_kmj) of the task to the VM; texec
kmjvcir , time to remotely execute the task on the VM instance; and trec kmjvcir , time to receive the output data
(_tkmj) of the task from the VM instance. The delay can be expressed by the following equation.
Cost Function
The model optimizes the total energy consumption when it is evaluated in RAE using a local private cloud. However, it optimizes the total energy consumption and the total monetary cost when it is evaluated in RAE using public clouds. The two cost functions considered in the model are explained as follows.
1) The first cost function, denoted as E(_) (3), represents the total energy consumption across all mobile devices whether executing their tasks locally on the mobile devices or remotely on the clouds.

$$\min\ [\ \pounds(\phi) + \alpha P(\phi)\ ]$$

To deal with the energy consumption and monetary cost in a single equation, we have introduced a parameter _ in the model. Depending upon the objective function of the cloud

users, _ can be set to different values. For example, if _ is set to zero, the model will minimize the total energy consumption
only without looking at the monetary cost. It is important to note that the value of _ is calculated as shown in Equation, and it depends on a tunable parameter wep and the ratio between the total energy consumption and the total monetary cost when tasks are offloaded using a centralized broker-node Without optimization.

The power rating of the WiFi radio of a mobile device in the send and receive states is estimated based on its current rating in these states and the voltage rating of the mobile device's battery. The current rating assumed during the send and receive states is 0:0857A and 0:0528A respectively [29], and the voltage rating of the mobile device's battery is 3:8V . It is also assumed that the power rating of the CPU in the compute state is higher than the send or receives state of the WiFi radio. An arbitrary value for the CPU power rating in the compute state is considered. Based on the above assumptions, the power ratings of WiFi radio and CPU in their respective states. Generally, resource intensive tasks are considered for offloading. The tasks considered for the model evaluation could be CPU, memory or I/O resource intensive, or a combination of these resources. However, we have characterized the tasks based on their local execution time. The value of the local execution time of a task could be due to CPU or memory intensiveness or both, or due to memory swapping among other simultaneously running tasks. When a task is executed locally on a mobile device, then the monetary cost is assumed to be zero and the type of resource intensiveness determines only the energy consumption. When a task is executed remotely on cloud resources, then the type of resource intensiveness determines the energy consumption and the monetary cost (if any).

VI. EXPERIMENTAL RESULTS

The proposed task scheduler model decides on task offloading is beneficial when the cost of offloading is less than the cost of executing the task on the mobile device. The cost of task offloading may include energy consumption, monetary cost, completion time of the task, etc. It is observed that the user-defined delay tolerance for an offloading task puts constraints on the remote completion time of the task. The data size of a task accounts for the energy consumption, monetary cost and the completion time of the task. When offloading a data intensive task, it may consume more energy than executing the task locally on the mobile device. On the other hand, offloading a task with small data size but with small delay tolerance may not be beneficial as well. Therefore, it is the combined effect of the delay tolerance and the data size of a task that influences task offloading decisions. Thus, every task may not be benefited from offloading; rather, it is a trade-off between the computation cost and the communication cost of the offloading task.

V. RELATED WORK

In general, task offloading with the aim to minimize the total energy consumption and the total monetary cost. In both RAEs, an overall improvement in energy consumption and/or monetary cost is observed when offloading using a centralized broker-node with optimization compared to offloading without optimization. Higher data rates are available while accessing resources from a local cloud than from public clouds. Thus, the energy consumption is high when offloading data intensive tasks onto public clouds compared to when offloading onto a local cloud. Therefore, offloading data intensive tasks may not be beneficial when using public clouds. The availability of higher data rates and lower network latency when using a local private cloud suggests that this environment is good for offloading realtime critical applications.

However, the downside of a local private cloud is that it may have limited resources. Overall, an improvement in energy saving when offloading to public clouds is more compared with offloading to a local private cloud. The difference in the overall improvement can be attributed to the effect of the availability of infinite and faster computation resources in the public clouds versus finite and slower computation resources in the local private cloud. In general, task offloading is beneficial when the cost of offloading is less than the cost of executing the task on the mobile device. The cost of task offloading may include energy consumption, monetary cost, completion time of the task, etc. It is observed that the user-defined delay tolerance for an offloading task puts constraints on the remote completion time of the task. The data size of a task accounts for the energy consumption, monetary cost and the completion time of the task. When offloading a data intensive task, it may consume more energy than executing the task locally on the mobile device. On the other hand, offloading a task with small data size but with small delay tolerance may not be beneficial as well. Therefore, it is the combined effect of the delay tolerance and the data size of a task that influences task offloading decisions. Thus, every task may not be benefited from offloading; rather, it is a trade-off between the computation cost and the communication cost of the offloading task.

VII. CONCLUSION

Our proposed system, a centralized broker-node based architecture was utilized to handle task scheduling on behalf of a large number of mobile devices. A general mathematical model for the centralized task scheduling problem was proposed with an aim to minimize the total energy consumption and the total monetary cost across all mobile devices of the system. The model was evaluated under two different resource augmentation environments for MCC, one using a local private cloud and the other using public clouds. The task scheduler model provided an optimal solution for the task scheduling problem (task assignment), and minimized the total energy consumption when evaluated in the local private cloud environment, and the total energy consumption and the total monetary cost when evaluated in public clouds environments, subject to various constraints.

The results showed that the total energy consumption and the total monetary cost across all mobile devices when offloading with optimization is less than when offloading without optimization using the centralized task scheduler.

FUTURE ENHANCEMENT

In this project, future work may include extending the scheduler model to consider network congestion, task priority and task execution redundancy while scheduling task offloading.

REFERENCES

[1] Manjinder Nir, And Marc St-Hilaire , "Economic and Energy Considerations for Resource Augmentation in Mobile Cloud Computing" Citation information: DOI 10.1109/TCC.2015.2469665, IEEE Transactions on Cloud Computing.

[2] M. Nir, A. Matrawy, and M. St-Hilaire, "Optimizing Energy Consumptionin Broker-Assisted Cyber Foraging Systems," in Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on. IEEE, 2014, pp. 576–583.

[3] M. Nir, A. Matrawy, and M. St-Hilaire, "An Energy Optimizing Scheduler for Mobile Cloud Computing Environments," in Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. IEEE, 2014, pp. 404–409.

[4] M. Nir and A. Matrawy, "Centralized Management of Scalable Cyber Foraging Systems," in Proceedings of the 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN), vol. 21. Elsevier, 2013, pp. 265–273.

[5] H. Wu, Q. Wang, and K. Wolter, "Tradeoff Between Performance Improvement and Energy Saving in Mobile Cloud Offloading Systems," in Proceedings of the Internatioanl Conference on Communications: Ist International workshop on Mobile Cloud Networking Services (MCN). IEEE, 2013, pp. 738–742.

[6] Y. Wen, W. Zhang, and H. Luo, "Energy-Optimal Mobile Application Execution: Taming Resource-Poor Mobile Devices with Cloud Clones," in INFOCOM, 2012 Proceedings. IEEE, 2012, pp. 2716–2720.