ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

EDTOS: Energy-Aware Dynamic Mobile Task Offloading and Scheduling Model in Mobile Cloud Computing

Erana Veerappa Dinesh S[1], Dr.Valarmathi K[2]

Assistant Professor (Ph.D. Scholar), Department of IT, V.P.M.M. Engineering College for Women, Krishnankoil[1]

Professor & Head, Department of ECE, P.S.R Engineering College, Sivakasi[2]

***Abstract*—Mobile devices play a significant role in the completion of the computational tasks in the day-to-day business activities. Offloading of tasks from the mobile devices to the cloud enhances the computational efficiency and battery lifetime of the mobile devices. But, the energy consumption during the communication activities should be balanced during the offloading process. The impact of the system parameters introduces additional challenge in the offloading performance. Hence, there is a need to make the offloading decision correctly based on the system parameters. The accurate offloading decisions are made based on the estimation of energy cost of the communication activities. An accurate offloading framework allows the mobile devices to obtain the correct decision based on the system parameters. This paper proposes an energy-aware dynamic mobile task offloading and scheduling model. In this work, an offloading framework is developed to enable the mobile devices to make the correct offloading decisions accurately. The framework accomplishes the correct decision after collecting all system parameters that have direct influence on the offloading decision. The proposed model achieved a significant reduction in the energy consumption and efficient scheduling of the tasks to the Virtual Machines (VMs) in the cloud.**

***Index Terms*—Energy-Aware Scheduling, Energy Cost, Genetic Algorithm (GA), Mobile Cloud Computing, Task Offloading, Task Scheduling**

I. Introduction

With the integration of mobile devices in the cloud computing environment, the Mobile Cloud Computing (MCC) is evolved. MCC extends the capabilities of the mobile devices and improves the experience of the user using the abundant cloud resources. With the utilization of clouds, the mobile applications will offload tasks to the clouds in the client-server model. Cloud computing is the one of the alternative ways to reduce the mobile task scheduling problems. Task offloading from the mobile devices to the cloud is essential to enhance their computing capabilities and energy-efficient operating systems, graphical user interfaces, communication protocols and task offloading [1]. Among them, task offloading offers promising results in the reduction of energy consumption in the mobile devices. Using the offloading approach, the mobile devices can easily offload the heavy tasks to the remote machines in the cloud and conserve the energy [2]. The energy consumption of the mobile devices can be reduced efficiently by offloading the heavy tasks from the devices to the cloud. The energy cost of the task offloading process is estimated and compared with the energy cost required for the local execution of the task. The energy consumption during the task offloading is caused due to the networking activities of the mobile device [3].

The offloading approach is classified based on the type of the remote machine such as web proxy, local powerful server and cloud. In the web proxy based offloading approach, the proxy serves as an intermediate between the mobile device and web server. In the server-based offloading, the server is located near to the mobile device. In the cloud-based offloading, the cloud provides the processing and storage resources to a mobile device. However, efficient task offloading requires better offloading decisions. An accurate offloading framework allows mobile devices to take the correct decision whether the task offloading is required or not based on the system parameters. In this work, we tackle this challenge by developing an offloading framework that allows the mobile devices to make the correct offloading decisions accurately. The framework accomplishes the correct decision after collecting all system parameters that have direct influence on the offloading decision. To consider this issue, we propose an Energy-aware Dynamic Mobile Task Offloading and Scheduling (EDTOS) model to enable a free flow communication between the cloud server and mobile device to reduce energy consumption.

The remaining sections of the paper are unfolded as follows: Section II reviews the current research works related to the task offloading process. Section III describes the proposed genetic algorithm based cloud task offloading and

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

scheduling approach. Section IV presents the performance analysis result of the proposed approach. Section V concludes the proposed work.

II. Related Art

The offloading approach is used for balancing the load in the mobile devices and improving the performance of the device by reducing the energy consumption. Kumar et al. [4]investigated an overview of the computational offloading approaches for the mobile systems. Liu et al. [5] presented an iterative decoupling algorithm for obtaining near-optimal offloading decisions to reduce the energy consumption of the mobile devices. Krishna et al. [6] introduced a model for task offloading using learning automata based decision making algorithm. Fahim et al. [7] adopted computational offloading of the tasks in a Mobile Device Cloud (MDC) environment. Altamimi and Naik [8] proposed a decision engine for offloading the tasks from the mobile devices to the cloud. Shi et al. [9] introduced effective allocation and scheduling of the task offloading requests for resolving the conflicts for the cloud resources. The cost for providing the cloud resources to the mobile devices was reduced and computational speed of the mobile device was improved. Dev et al. [10] proposed an analytical model of end-to-end energy consumption for task offloading and evaluated the theoretical limits of the offloading. Cao et al. [11] proposed a combined framework for offloading task and data of the mobile devices. The proposed framework achieved the reduction in the energy consumption and data usage. Zhang et al. [12] studied the scheduling policy for the combined execution of the tasks in the MCC. The energy consumed by the mobile device was reduced and time deadline constraints were satisfied. Zhou et al. [13] presented a cuckoo-based flexible scheme for offloading the computational-intensive task in the MCC environment. The energy consumption was reduced and performance of the mobile devices was improved.

Mtibaa et al. [14] developed computational offloading scheme for increasing the lifetime of the MDC and used a testing platform for evaluating the computational models. Lin et al. [15] investigated the scheduling problems in the MCC environment and proposed a scheduling algorithm for migrating the tasks of the mobile devices onto the cloud. Zhou et al. [16] proposed a context-aware algorithm to make the code offloading decisions and select the appropriate wireless medium and cloud resources. The performance of the MCC was improved by selecting the suitable resources based on context of the mobile devices. Xia et al. [17] considered a problem of location-aware offloading in a two-tiered mobile cloud computing environment comprising a local cloudlet and remote clouds. An effective algorithm was devised for fair sharing of the cloudlet by consuming proportionate amount of energy from the mobile device. Jia et al. [18] presented an algorithm for the online offloading of tasks to reduce the time taken for completing the execution of the application on the mobile device. A load-balancing heuristic was used for offloading the tasks to the cloud and improve the parallelism between the mobile device and cloud. Chen et al. [19] proposed an opportunistic task scheduling to achieve flexible cost-delay tradeoff between the remote cloud and mobile cloudlets service modes. The offloading approach may reduce energy consumption and improve the performance of the mobile devices. But, it depends on the parameters such as the network bandwidth and the data exchange rate. The decisions are usually made by analyzing parameters such as network bandwidth, operating speed, load and available memory space of the cloud server, and the amount of data exchanged between the cloud servers and mobile devices.

III. Proposed Work

The Genetic Algorithm (GA) is an adaptive heuristic search technique used for solving the constrained and unconstrained optimization problems. It modifies a population of the individual solutions in a repeated way. The individuals are selected from the current population at a random manner. The selected individuals are used to produce the children for the next generation. Over the succeeding generations, the population progresses to an optimal solution.
The three main rules of the GA are

- Selection
- Crossover
- Mutation

The general procedure of the GA algorithm is described below

GA Algorithm

```
Choose initial population
Evaluate fitness value of each individual
Compute average fitness of population
Do
    Select the best-ranking individuals for reproduction
    Mate the pair of individuals at random
    Apply the crossover and mutation operators
    Estimate the fitness value of each individual
    Determine the average fitness value of population
Until terminating condition is satisfied
```

The main aim of the proposed approach is to assign each task to a Virtual Machine (VM), such that all the tasks are completed with minimum energy consumption. The energy consumption of each VM after allocating the task to the VM is calculated as

$$E_{ij}=p_i\, t_i \quad (1)$$

Where p_i is the power consumed by each task allocated to the VM and t_i is the execution time of the task. The total amount of energy consumed by the VM is denoted as τ_{ij}.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

In the proposed model, the cloud application involves a collection of jobs that perform complex tasks using the cloud resources. Generally, a set of applications A={a_1,… …,a_n} arrive at a certain time period. During the scheduling process, the client submits a service request for the application to the Cloud Service Provider (CSP). The service request is forwarded to the Cloud Service Manager and scheduled to the VMs using the energy-aware task scheduler. The CSP provides the requests submitted by the users with different resource requirements. In the next step, the Cloud Service Manager will analyze the resource requirements of each granularity and mapping it on to optimal VMs to reach an effective solution. The Cloud Task Manager manages the status of the task, determines the scheduling sequence and allocates each job to suitable VMs using the scheduler.

The GA algorithm starts with the initialization of population for defining the set of chromosomes that represents the possible solution to the task scheduling problem. During the scheduling process, each chromosome represents a sequence of schedules. Each gene in the chromosome represents the energy consumption of a task. The value of the gene is 0, if the energy consumed by the task is greater than τ_{ij} . The value of the gene is 1, if the energy consumed by the task is less than τ_{ij} . The chromosome having best fitness value is selected for producing the next generation. Fig.1 shows the flow diagram of the proposed EDTOS model. Fig.2 illustrates the proposed model.

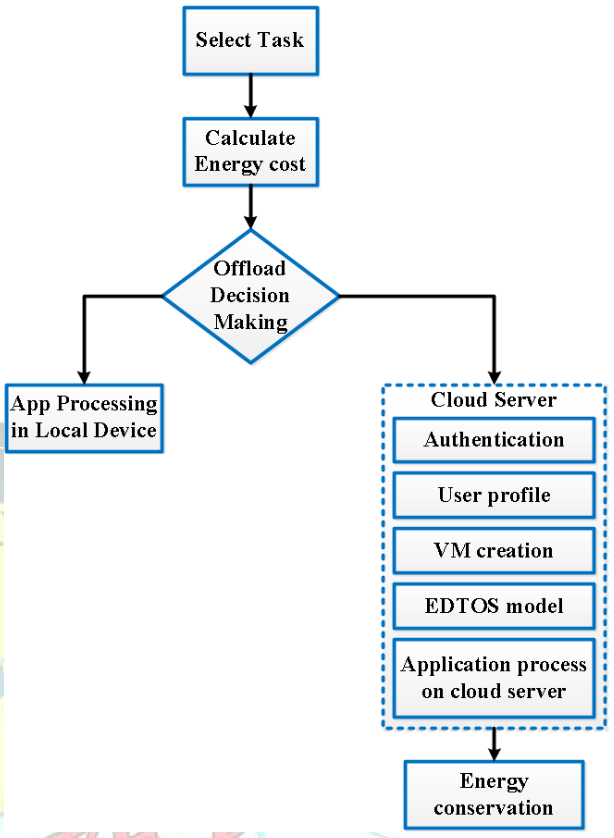


Fig.1 Flow diagram of the proposed EDTOS model

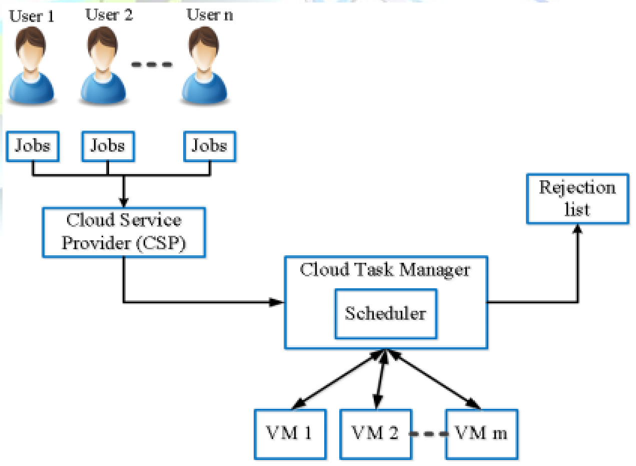


Fig.2 Proposed Model

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

Initially,a job is received from the user. The job execution request is initialized with the deadline and number of VMs required to process the jobs. The received jobs are directed to the CSP. All tasks or requests are submitted to the CSP. The size of the task and execution time for each job (i.e. Energy Value) are computed. The energy value of each job is calculated by using the formula $E_{ij} = p_i t_i$. If the energy value is greater than the predefined threshold value τ_{ij}, the task is moved to the CTM. The CTM monitors the jobs and sends all the information to the Scheduler. Then, the scheduler performs scheduling of the job by checking the inequality constraints and arranging the jobs. The scheduler applies improved GA and decides appropriate task to be allocated to the VMs in the host. If the task cannot be completed within its deadline, the task is moved to the rejection list.

Task Scheduling Algorithm

Step 1: Receives a job from the user
Step 2: Initialize the request with deadline and number of VMs required to process the jobs
Step 3: The received jobs will be directed to the CSP. All tasks or requests are submitted to the CSP.
Step 4: Compute the size of the task and execution time for each job (i.e. Energy Value). Calculate Energy value of each job by using the formula $E_{ij}=p_i t_i$.
Step 5: If Energy value is greater than the predefined threshold value τ_{ij} , it moves to CTM.
Step 6: CTM monitors the jobs and sends all the information to Scheduler.
Step 7: Schedule the job by checking the inequality constraints and arrange the jobs accordingly.
Step 8: Scheduler applies improved GA and decides the allocation of appropriate task to the VMs in the host.
Step 9: If the task cannot be completed within the deadline, the task is moved to the reject list.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

IV. Performance Analysis

Table I depicts the simulation parameters for each task. Table II shows the time spent in Idle/Active Mode for each storage and Table III presents the energy consumption in idle/active mode for one task. Table IV and Table V presents the time spent and energy consumption in the Idle/Active Mode for 10 tasks.

Table I Simulation Parameters for each task

Power consumption	11.27 Watt
Queue waiting time	0.000 seconds
Transaction time	0.019187 seconds
Energy consumption	0.216 Joules

Table II Time Spent in Idle/Active Mode for each storage

Idle interval	0.5 seconds
Time in Idle mode	0.5 seconds
Time in Active mode	0.019 seconds
Simulation time	0.519 seconds

Table III Energy Consumption in Idle/Active Mode

Energy consumed in Idle mode	3.450 Joules
Energy consumed in Active mode	0.216 Joules
Total energy consumption	3.666 Joules

Table IV Time Spent in Idle/Active Mode for 10 tasks

Time in Idle mode	0.623 seconds
Time in Active mode	0.324 seconds
Simulation time	0.947 seconds

Table V Energy Consumption in Idle/Active Mode for 10 tasks

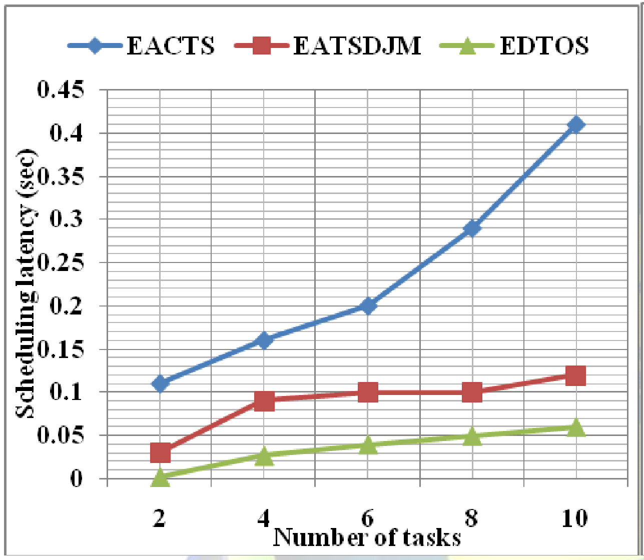
Energy consumed in Idle mode	4.3 Joules
Energy consumed in Active mode	3.649 Joules
Total energy consumption	7.949 Joules

Table VI Power, time and energy consumption analysis

Cloudlet	**Power consumption (W)**	**Idle Interval (sec)**	**Transaction time (sec)**	**Energy consumption (Joule)**
1	11.270	0.100	0.018062	0.204
2	11.270	0.082	0.022270	0.251
3	11.270	0.078	0.022750	0.256
4	11.270	0.077	0.043482	0.490
5	11.270	0.057	0.028006	0.316
6	11.270	0.072	0.038395	0.433
7	11.270	0.062	0.040788	0.460
8	11.270	0.059	0.063015	0.710
9	11.270	0.037	0.046970	0.529

The proposed EDTOS model is compared with the existing Energy Aware Task Scheduling and Dynamic Job Mapping (EATSDJM) [20] and Energy-Aware Communication and Task Scheduling (EACTS) technique [21]. The number of tasks vary from 2-10.

Fig.3 depicts the scheduling latency of the proposed EDTOS model and existing EACTS and EATSDJM techniques. There is a gradual increase in the scheduling latency with the increase in the number of tasks. Due to the dynamic job mapping, the EATSDJM technique requires minimum latency than the EACTS technique. The scheduling latency of the proposed EDTOS model is found to be minimum than the existing EACTS and EATSDJM techniques. The proposed EDTOS model yields minimum scheduling latency of about 85.36% and 50% than the EACTS and EATSDJM techniques. Fig.4 shows the average energy consumption of the proposed EDTOS model and existing EACTS and EATSDJM techniques. With the increase in the number of tasks, there is a linear increase in the energy consumption. However, due to dynamic job mapping, the EATSDJM technique requires minimum energy consumption when compared to EACTS. The energy consumption of the proposed EDTOS model is about 64.73% and 59.31% lesser than the EACTS and EATSDJM techniques. Fig.5 illustrates the execution time analysis of the proposed EDTOS model and existing EACTS and EATSDJM techniques. The proposed EDTOS model requires minimum execution time than the existing EACTS and EATSDJM techniques. The execution time of the proposed EDTOS model is 81.77% and 64.35% lesser than the EACTS and EATSDJM techniques. Hence, our proposed EDTOS model is efficient that the EACTS and EATSDJM techniques.



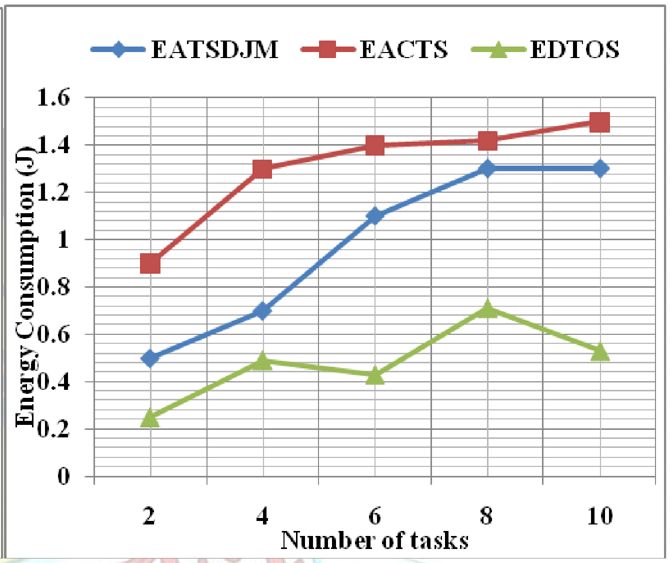


Fig.3 Scheduling Latency of the proposed EDTOS model and existing EACTS and EATSDJM techniques. Fig.4 Energy consumption analysis of the proposed EDTOS model and existing EACTS and EATSDJM techniques

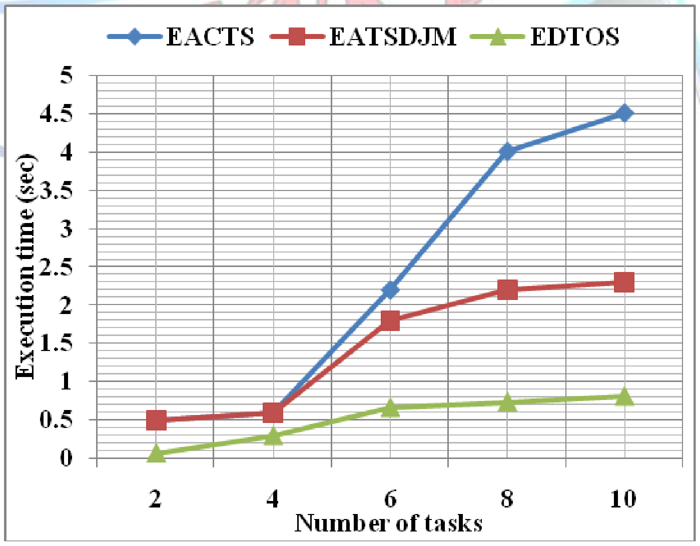


Fig.5 Execution time analysis of the proposed EDTOS model and existing EACTS and EATSDJM techniques

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

V. CONCLUSION

Task offloading to the cloud improves the operating performance of the mobile device by overwhelming the constraints such as limited battery energy, processing capability and memory capacity. But, there is a need to measure the energy consumed by the task, so as to increase the offloading efficiency. This paper presented the energy-aware dynamic model for offloading and scheduling the tasks in the cloud computing environment. The energy value of the job is calculated based on the size of the task and execution time of each job. If the energy value is higher than the threshold value, the job is moved to the cloud task manager. The improved GA is applied for scheduling the tasks to the VMs. The experimental results denote that the proposed EDTOS model requires minimum scheduling latency, energy consumption and execution time than the existing techniques, due to the dynamic offloading and scheduling of the tasks.

REFERENCES

[1] K. Naik, *A survey of software based energy saving methodologies for handheld wireless communication devices*: Department of Electrical and Computer Engineering, University of Waterloo, 2010.

[2] W. Zhang, Y. Wen, J. Wu, and H. Li, "Toward a unified elastic computing platform for smartphones with cloud support," *IEEE Network,* vol. 27, pp. 34-40, 2013.

[3] M. Altamimi, A. Abdrabou, K. Naik, and A. Nayak, "Energy cost models of smartphones for task offloading to the cloud," *IEEE Transactions on Emerging Topics in Computing,* vol. 3, pp. 384-398, 2015.

[4] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications,* vol. 18, pp. 129-140, 2013.

[5] K. Liu, J. Peng, H. Li, X. Zhang, and W. Liu, "Multi-device task offloading with time-constraints for energy efficiency in mobile cloud computing," *Future Generation Computer Systems,* vol. 64, pp. 1-14, 2016.

[6] P. V. Krishna, S. Misra, D. Nagaraju, V. Saritha, and M. S. Obaidat, "Learning automata based decision making algorithm for task offloading in mobile cloud," in *Computer, Information and Telecommunication Systems (CITS), 2016 International Conference on*, 2016, pp. 1-6.

[7] A. Fahim, A. Mtibaa, and K. A. Harras, "Making the case for computational offloading in mobile device clouds," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 203-205.

[8] M. Altamimi and K. Naik, "A Practical Task Offloading Decision Engine for Mobile Devices to Use Energy-as-a-Service (EaaS)," in *2014 IEEE World Congress on Services*, 2014, pp. 452-453.

[9] C. Shi, K. Habak, P. Pandurangan, M. Ammar, M. Naik, and E. Zegura, "Cosmos: computation offloading as a service for mobile devices," in *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*, 2014, pp. 287-296.

[10] V. C. Dev, V. Namboodiri, M. Tandel, and V. Venkitachalam, "Is there a Net Energy Saving by the use of Task Offloading in Mobile Devices?," 2015.

[11] Y. Cao, C. Long, T. Jiang, and S. Mao, "Share communication and computation resources on mobile devices: a social awareness perspective," *IEEE Wireless Communications,* vol. 23, pp. 52-59, 2016.

[12] W. Zhang, Y. Wen, and D. O. Wu, "Energy-efficient scheduling policy for collaborative execution in mobile cloud computing," in *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 190-194.

[13] Z. Zhou, H. Zhang, L. Ye, and X. Du, "Cuckoo: flexible compute-intensive task offloading in mobile cloud computing," *Wireless Communications and Mobile Computing,* 2016.

[14] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," in *ACM SIGCOMM Computer Communication Review*, 2013, pp. 51-56.

[15] X. Lin, Y. Wang, Q. Xie, and M. Pedram, "Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment," *IEEE Transactions on Services Computing,* vol. 8, pp. 175-186, 2015.

[16] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "A context sensitive offloading scheme for mobile cloud computing service," in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 869-876.

[17] Q. Xia, W. Liang, Z. Xu, and B. Zhou, "Online algorithms for location-aware task offloading in two-tiered mobile cloud environments," in *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 2014, pp. 109-116.

[18] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 352-357.

[19] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, and K. Hwang, "Opportunistic Task Scheduling over Co-Located Clouds in Mobile Environment," *IEEE Transactions on Services Computing,* 2015.

[20] J. Kalaivani and B. Vinayagasundaram, "Energy aware scheduling and dynamic job mapping algorithm for Network-on-Chip architectures," in *International Conference on Recent Trends in Information Technology (ICRTIT),*, 2016, pp. 1-4.

[21] J. Hu and R. Marculescu, "Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, 2004, pp. 234-239.