International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P), Volume-5, Issue-2, June 2016

An Optimal Partition Approach for Mobile to Cloud Offloading

Pummy Dhiman, Kamal Kumar

Abstract— **Now a days smartphones have become an essential part of our life as the most convenient communication tool. The main point of attraction of them is feature rich applications. These applications imply a heavy workload on processors, which causes a significant energy cost as they are battery powered. In this paper, we represent a computation offloading scheme on mobile devices to improve the performance and energy consumption. When a task comes to the mobile device, this scheme partition the task into a client–server distributed program, such that the client code runs on the handheld device and the server code runs on the server. We give optimal partition algorithm to find an optimal program partition for given program input data. Experimental results show significant improvement of performance and energy consumption on an android handheld device through computation offloading.**

Index Terms— **Application Partitioning, Computation Offloading, Mobile Cloud Computing, Virtual Machine.**

I. INTRODUCTION

The full potential for smartphones and tablet PCs may be constrained by certain technical limits such as battery endurance, computational performance, and portability. Modern mobile applications own more powerful functions but need larger computation, which consume more battery energy.It is common practice for mobile devices to offload computationally heavy tasks to a cloud, which has greater computational resources and receive the results from cloud.By offloading computation to resource rich Cloud, energy consumption on the mobile device can be saved considerably and limitations of mobile devices can be overcome, this type of offloading is known as Computation Offloading.

This paper represents a computation offloading scheme on handheld devices to improve the performance and energy consumption. We give optimal partition algorithm to find an optimal program partition for given program input data. Experimental results show significant improvement of performance and energy consumption on an android handheld device through computation offloading.

Pummy Dhiman, Department of Computer Science & Engineering, Kurukshetra University, Kurukshetra, SDDIET,Panchkula,India.

Kamal Kumar, Department of Computer Science & Engineering, Kurukshetra University, Kurukshetra, SDDIET, Panchkula, India.

II. COMPUTATION OFFLOADING SCHEME

In our computation offloading scheme we divides the whole program computation into tasks. We divide the tasks into server tasks and client tasks such that server tasks run on the server and client tasks run on the handheld device. The terms *cyber foraging* and *surrogate computing* are also used to describe computation offloading.[11]

III. RELATED WORK

Li.Z, Wang.C, Xu.R[1] presents an offloading scheme based on profiling information about computation time and data sharing at the level of procedure calls. A cost graph is constructed and a branch-and-bound algorithm is applied to minimize the total energy consumption of computation and the total data communication cost. **Wang.C, Li.Z [2]** presents a computation offloading scheme on mobile devices and proposes a polynomial time algorithm to find an optimal program partition. Both the static information and the run-time information were used. A heuristic algorithm is applied into the clusters to find the optimal partition to minimize the execution cost of the program. **Kumar.K, Lu.Y[3]** suggest a program partitioning based on the estimation of the energy consumption (communication energy and computation energy) before the program execution. **Carroll.A, Heiser.G [4]** Good energy management requires a good understanding of where and how the energy is used. They derive an overall energy model of the device as a function of the main usage scenarios. **B. Chun, S. Ihm [5]** CloneCloud is a combination of static analysis and dynamic profiling. It partition applications automatically at a fine granularity while reducing execution time and energy use for a target computation and communication environment. History based profiling is used. But real Network and device conditions cannot be generalized. **X. Feng, D. Fangwei [7]** Phone2Cloud, a computation offloading-based system is devised for energy saving on smartphones in the context of mobile cloud computing. This system is a semi-automatic offloading system. **Flores. H, Srirama. S [8]** compared the mobile cloud computing models for offloading and delegation. Mobile applications may be bonded to cloud resources by following a task delegation or code offloading criteria. **Roopali, Kumari.R [9]** presented the challenges with offloading such as latency rate which mainly depends on factors like code to be offloaded, distance between smart phone device and the remote server, network bandwidth, heterogeneous environments and results of computations. **Wang.Y and Chen.I [10]** They note that trust management to support mobile cloud computing is a totally unexplored area, especially for an ad hoc mobile cloud comprising mobile nodes as resource providers without involving a remote cloud. **Kaur.K and Kaur.P [12]** focuses on an emerging technology known as mobile offloading and mashup. The offloading uses the quality of services concept to remove mobile application limitations. Cloud services are the application programming platform where users can create new applications and mashup their functionalities. **Wang, Yanzhi, Lin.X[14]** Computation offloading, as one of the main advantages of MCC, is a paradigm/solution to improve the capability of mobile services through migrating heavy computation tasks to powerful servers in clouds. Computation offloading yields saving energy for mobile devices when running intensive computational services, which typically deplete a device's battery when are run locally. **Liu.J, Ahmed.E, Shiraz.M[15]** They consider application partitioning to be an independent aspect of dynamic computational offloading and also review the current status of application partitioning algorithms (APAs) to identify the issues and challenges.

IV. OFFLOADING DECISION MAKING ALGORITHM

This optimization partitions mobile applications, so that their energy intensive functionality is executed in the cloud, without draining the battery. In this paper, we present a novel offloading approach that combines the advantages of the prior state of the art both in partitioning mobile applications and in dynamically adapting mobile execution targets in response to fluctuations in network conditions. Cloud offloading is a mobile application optimization technique that makes it possible to execute the application's energy intensive functionality in the cloud, without draining the mobile device's battery.[13] We use EM algorithm for optimally partition the mobile application.

EM Algorithm
"EM Algorithm" where EM stands for Expectation and Maximization is used in our research to make offloading decision.

$$P(\text{Model} \mid \text{Data}) = \frac{P(\text{Data} \mid \text{Model})P(\text{Model})}{P(\text{Data})}$$

Likelihood Function

Find the "best" model which has generated the data.The EM algorithm requires us to iterate through the following two steps:

- **The Expectation Step:** Using the current best guess for the parameters of the data model, we construct an expression for the log-likelihood for all data, observed and unobserved, and, then, marginalize the expression with respect to the unobserved data.
- **The Maximization Step:** Given the expression resulting from the previous step, for the next guess we choose those values for the model parameters that maximize the expectation expression. These constitute our best new guess for the model parameters.

Following flowchart explains the procedure of computation offloading:

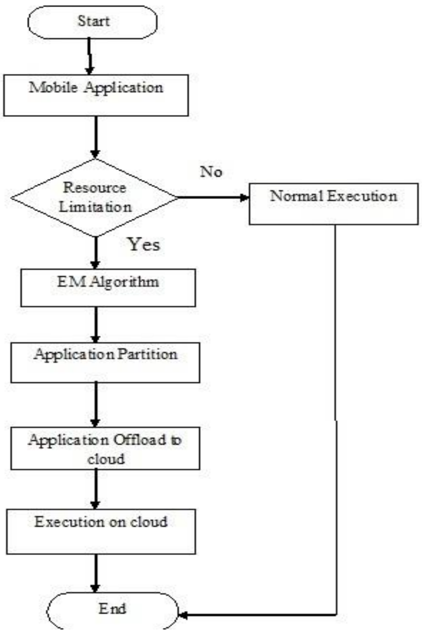

Fig. 1: Flowchart

Fig. 1 shows the general flow of operations involving duration of the application partitioning and computation offloading. The application does not demand huge resources only then it executes normally on mobile and the application profiling mechanism evaluates computing resources utilization, availability of resources and computing requirements of the mobile application. In the critical situation when there is insufficient resources on the smartphone, optimal partition approach is activated to separate the offloadable part from the non-offloadable part. Upon successful execution of the remote components of the application, the result is returned to the main application running on the phone.

V. PERFORMANCE METRICS

In this paper, we investigate modeling and optimality by considering three parameters: the energy consumption on the mobile device, the application response time and cost. The performance of an offloaded task is judged based on the goals set by the user.

VI. RESEARCH METHODOLOGY

The offloading decision is fully automatic by adaptive partition of task using EM(Expectation Maximization) algorithm and analysis the response time, energy and cost. It has been found that in previous work

- process is offload manual by user which is automatic,
- Offloading decision depend on the previous task, that's why maintain the previous task information which take more time for processing.
- predict the offloading which is depend on the predict in model and it give some time error when take wrong prediction.
- maximum offloading decision is static is not depend on the task .

We deploy three models- cost model, energy model and weighted model.

Cost Model In this model, we model the cost as the program response time and calculate the memory gain of offloading.

Energy Model In this model, we calculate the energy gain of offloading. Energy is the total amount of work performed over a period of time. We also calculate the time of offloading of given tasks. After this calculation we perform offloading.

Weighted Model In this model we have the memory gain on the cloud and mobile, which is not available in previous models. Then the EM algorithm is called to partition the task optimally.

Algorithm – weighted model
for (i=1; i< k; i++)
{
Calculate the memory gain of offloading
gain $\leftarrow \sum_{i=1}^{n}$ (mobile $_i$ – cloud $_i$) – input $_i$ – return $_i$
end
$U_i \leftarrow$ max { gain $_{i\ to\ i}$ $1 < i \leq k$ }
Integrating Point $\leftarrow$ EM (max (gain $_{i\ to\ i}$ $1<i\leq k$))
for (j=2; j< k; j++);
{
calculate the time of offloading of j^{th} task
$X_j = x_{j-1} - (mobile_{j-1} - cloud_{j-1}) + input_{j-1} - input_j$
}
Offloading point

VII. EXPERIMENT

We use android handheld device and the server is a Dell having Intel(R) Core(TM) 2 Duo CPU 2.10 GHz. The CloudSim toolkit is chosen as a simulation platform, as it is a modern simulation framework aimed at Cloud computing environments.[6] The no of nodes are used 5-500, 45 virtual machines are used each having 500MB RAM and 5 MIPS processor frequency.

VIII. RESULT

As per earlier discussion, simulation is performed using cloudSim simulator to analyze the effect of computation offloading. Here, this performance is evaluated based on different performance metrics like cost, response time and energy consumption.

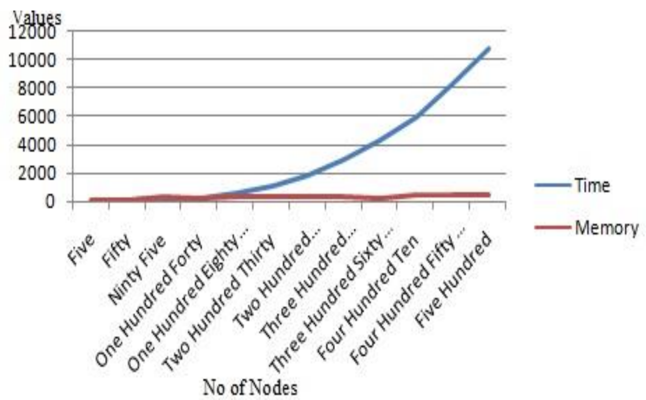

Fig.2 Time and Memory used in cost model

Fig.2 shows the simulation graph between response time and memory, that we consider in our cost model.

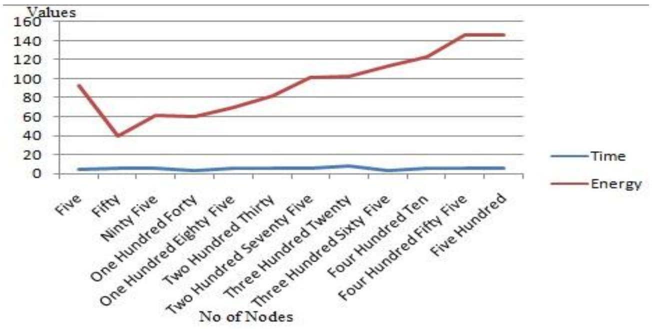

Fig. 3 Simulation graph for Energy Model

The energy consumption parameter represents the battery power consumed by the mobile device.

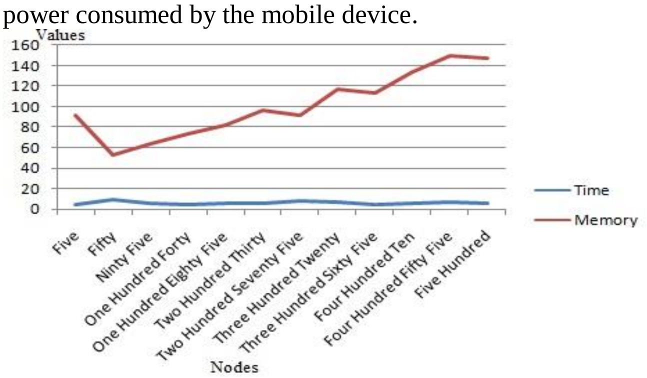

Fig.4 Simulation graph for Weighted Model

Fig. 4 shows the time and memory used by task after optimal partitioning for computation offloading, and it clear that it uses less time and memory as compare to the other models.

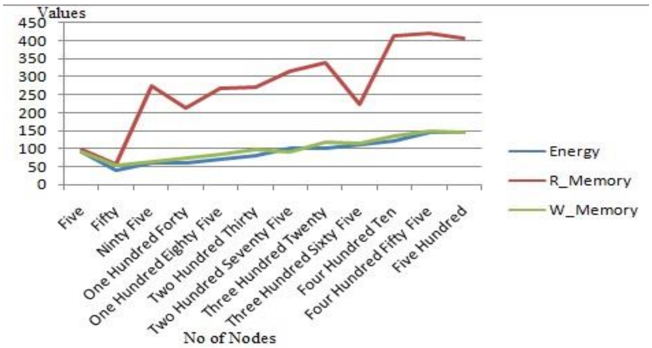

Fig. 5 Energy used in cost model and weighted model

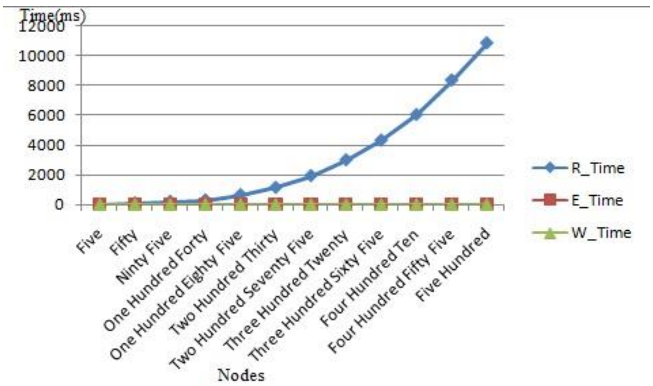

Fig. 5 Time estimation in all models

Fig.5 shows the cost simulation graph for all the three models. It is the comparison of the response time of the all three models.

IX. CONCLUSION AND FUTURE SCOPE

In this paper, we have presented a computation offloading scheme on handheld devices. Our partition algorithm finds optimal program partitioning for given program input data. Experimental results show that our computation offloading scheme can significantly improve the performance and energy consumption on handheld devices. However, we believe our computation offloading scheme can still be improved. More work is needed to be done for offloading field. Future work can be extended in the following fields: Here, we use EM algorithm for optimal partition the application running on smartphone, in future we will use metaheuristic algorithm, since metaheuristics are one of the most practical approaches for solving hard optimization problems. This approach is interesting for very large problem instances. We will also take our operations to hardware for better understanding the calculate delay and memory usage.

ACKNOWLEDGEMENT

I consider it is a privilege to express my gratitude and respect to all those who guided me in completion of technical paper. I would like to express my gratitude to **Mrs. Nidhi Sharma,** Hod, Department of Computer Science and Engineering, Swami Devi Dyal Institute of Engineering & Technology, Barwala, Panchkula, for providing me with all the necessary facilities for the research.I am also extremely indebted to my guide **Mr. Kamal Kumar,** Assistant Professor. This work would not have been possible without his guidance, support and encouragement. Under his guidance I successfully overcame many difficulties and learned a lot.

I would like to thank to other faculty members of the CSE Department and my friends for their intellectual support throughout my research work.

REFERENCES

[1] Li Z,Wang C,Xu R,"Computation offloading to save energy on handheld devices: a partition scheme",In Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems (CASES),238–246,2001

[2] ChengWang and Zhiyuan Li,"A computation offloading scheme on handheld devices", Journal of Parallel and Distributed Computing,2003

[3] Karthik Kumar and Yung-Hsiang Lu,"Cloud computing for mobile users: Can offloading computation save energy?",Computer, vol. 43, no. 4, pp. 51–56, IEEE 2010

[4] Aaron Carroll and Gernot Heiser,"An Analysis of Power Consumption in a Smartphone", USENIXATC'10 Proceedings of the 2010 USENIX conference on USENIX annual technical conference,USENIX Association Berkeley, CA, USA ,23 June 2010

[5] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in Proc. ACM The European Professional Society on Computer Systems (EuroSys'11), Salzburg, Austria,pp. 301–314, April 2011

[6] R. N. Calheiros, R. Ranjan, A Beloglazv, C. A. F. De Rose and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", in Software-Practice and Experience,vol. 41, pp. 23-50, 2011

[7] X. Feng, D. Fangwei, L. Jie, K. Xianglie, Y.Laurence and M. Jianhua, "Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing" Springer Issue on Information Systems Frontiers, October 20, 2013.

[8] Huber Flores, Satish Narayana Srirama, Rajkumar Buyya,"Computational Offloading or Data Binding? Bridging the Cloud Infrastructure to the Proximity of the Mobile User"IEEE,2014

[9] Roopali, Rajkumari,"Overview of Offloading in Smart Mobile Devices for Mobile Cloud Computing",Vol 5, Number 6, 2014

[10] Yating Wang and Ing-Ray Chen, "A Survey of Mobile Cloud Computing Applications: Perspectives and Challenges", Volume 80 Issue 4, February 2015.

[11] Karthik Kumar·Jibang Liu·Yung-Hsiang Lu· Bharat Bhargava,"A Survey of Computation Offloading for Mobile Systems", Springer Issue on Mobile Network Application,10 April, 2012.

[12] Kaur.K and Kaur.P," Computational Offloading for Mashup Services in Mobile Cloud Computing Environment", International Journal of Grid Distribution Computing Vol. 8, No. 3, pp.145-154,(2015)

[13] Kwon.Y and Tilevich.E," Reducing the Energy Consumption of Mobile Applications Behind the Scenes",

[14] Wang, Yanzhi, Xue Lin, and Massoud Pedram. "A Bayesian game formulation of power dissipation and response time minimization in a mobile cloud computing system." Proc. IEEE Mobile Service, pp. 7-14, 2013.

[15] Liu.J, Ahmed.E, Shiraz.M, Gani.A, Buyya.R and Qureshi.A ,"Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions", Journal of Network and Computer Applications 48 (2015) 99–117

Pummy Dhiman is currently pursuing masters in Computer Science Engineering, in SDDIET, KUK University, India

Kamal Kumar is currently Assistant Professor in Computer Science Engineering, in SDDIET, KUK University, India.