Dynamic Computational Offloading To Mobile Cloud Extend Lifetime in Mobiles

T. Sathish Kumar[1] M. Melowin Mary[2] S. Bhavapriya[3] N. Mohanambal[4] B. Monica[5]
[1,2,3,4,5]Saranathan College of Engineering

Abstract— Offloading is a hot topic which is quickly taking the world by storm since mobile phones are exploding and penetrating very quickly. But some drawbacks include the battery lifetime which is fast depleting. To overcome this lots of solutions are proposed. Some solutions are very hardware oriented while some are software oriented. To improve battery lifetime has become the most desired study in mobile commuting in recent times. Nowadays mobile apps decrease the lifetime of the smart phone battery resulting in loss of work time or data. The proposed model suggests and implements an offloading strategy to cloud where the data is processed and then return the result. The result is processed quickly and also the energy drain is very less, thereby extending the lifetime. Other resources like extending the mobile phone lifetime, components, saving RAM and processor overloading including hanging are also handled.

Key words: Mobile Computation, Offloading, Cloud Computing

I. Introduction

The use of computation offloading in mobile systems is rapidly increasing where mobile user energy consumption is improved by offloading job execution to remote cloud servers, rather than performing the computations locally. It has been shown that remote application execution can significantly improve mobile battery lifetime in these types of situations. Computation offloading exploits the use of cloud-based servers that have significantly more resources than that of a typical mobile user. There are also studies that have considered computation offloading from an application execution viewpoint, where jobs are partitioned into multiple local and remote execution components.

The main issue with computation offloading in MCC architectures is the requirement of pre-installed mobile services in the servers. In VMM based communication models, the memory image of a central cloud server is migrated in the edge servers which lowers the communication cost and overall bandwidth utilization in highly dense environments. Offloading is done to save energy and extend the lifetime of the device using it. This enables the lifetime of the battery to be extended

II. Related work

Khan in 2015 in the paper “A survey of computation offloading strategies for performance improvement of applications running on mobile devices proposed clientserver settings, virtual machine migration (VMM), and mobile agent configurations where the client-server based settings the computation offloading components resides on the mobile device and offloads the computations after checking the cost benefit analysis of offloading the work. J. Liu et al in Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions” proposed application partitioning and resource-hungry computational tasks are offloaded to the nearest designated surrogate cloud servers.

E. E. Marinelli, in “Hyrax: cloud computing on mobile devices using mapreduce. R. W. Ahmad, er al in A survey on virtual machine migration and server consolidation frameworks for cloud data centers,” proposed VMM which introduces the communication models, the application clones are migrated in cloud environments to augment the mobile devices with cloud resources. However, the management of mobile agents and security of clones are the main issues in mobile agent based MCC environments.

S. Abolfazli, et al in Cloud based augmentation for mobile devices: motivation, taxonomies, and open challenges proposed computation offloading schemes work with single-site and multiple-site surrogate settings and single site surrogates, the application components are offloaded into the same server in the MCC architecture while the multiple-site surrogates facilitate in two ways.

III. Architecture

The architecture accepts jobs for processing and sends the jobs to the cloud for processing. This reduces the time and saves energy while parameters like heat, temperature and processor are monitored.

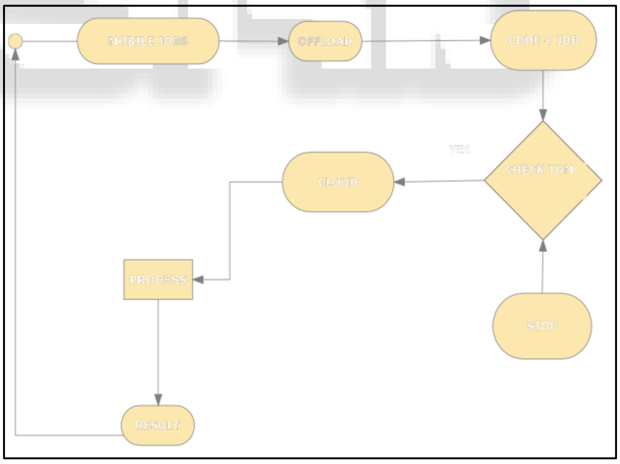


Fig. 1: Proposed Offloading Model

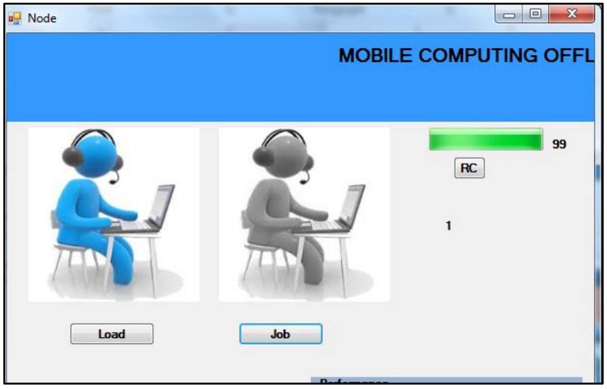


Fig. 2: Existing Implementation

IV. Proposed Model

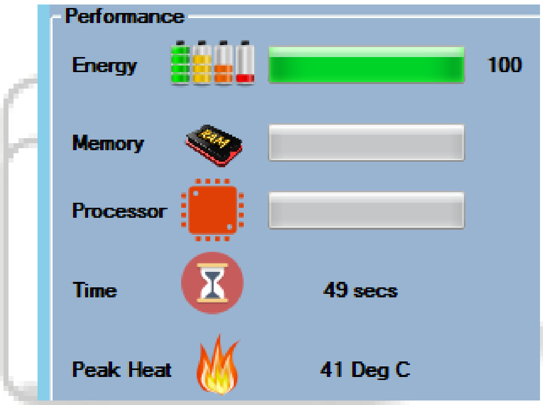
The proposed model uploads the job data to the mobile cloud and then the cloud processes the job by sending back the result. The system which is a framework pushes the data to the handset asynchronously. So Mobile Tasks are delegated to the Cloud and hence energy is saved.

V. Implementation

The model implemented monitors several parameters like energy, processor, memory, heat, and time etc for the job or tasks processed in the cloud. A graph is drawn to show the difference between existing and proposed parameters.

Image jobs are taken as input and processed like converting to grayscale, doing twirl operations, segmentation process and wave patterns. Image processing is chosen because that causes huge memory to be taken up than normal text operations. Also the input is an image file while the file is transmitted to the cloud and the result file is obtained after processing from the cloud to the mobile client.

The entire process is automated and the generated client job is offloaded to the cloud and the processed job is sent back to the client.



Performance

Energy 100

Memory

Processor

Time 49 secs

Peak Heat 41 Deg C

Fig. 3:

VI. Evaluation

The evaluation of the proposed and existing models show that the proposed model occupies less energy than the existing model, apart from occupying less memory storage, less cpu and time.

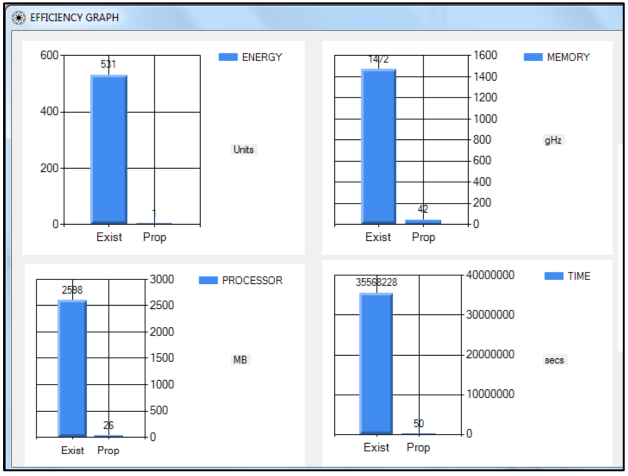


Fig. 4: Graphical Output 1

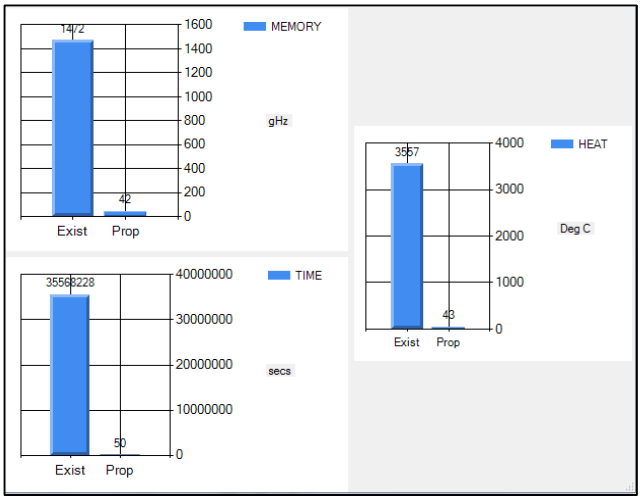


Fig. 5: Graphical Output 2

VII. Conclusion

The model thus proves that offloading of computation to mobile cloud is a popular and future task for applications in the mobile environment. This model proves that computation offloading is here to stay and is a good model for extending the battery lifetime and energy of the mobile. As part of future enhancements dedicated channels may be setup for energy and computation offloading of specific apps.

References

[1] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, “A Game Theoretic Resource Allocation for Overall Energy Minimization in Mobile Cloud Computing System,” in ISLPED, ser. ISLPED ’12. ACM, July 2012, pp. 279–284.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the Art of Virtualization,” SIGOPS Oper. Syst. Rev., vol. 37, no. 5, pp. 164–177, 2003.

[3] [Online]. Available: http ://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white paper c11-520862.

[4] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, “Gearing Resource-Poor Mobile Devices with Powerful Clouds: Architectures, Challenges, and Applications,” IEEE Wireless Communications, vol. 20, no. 3, pp. 14–22, Jun. 2013.

[5] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, “Challenges on Wireless Heterogeneous Networks for Mobile Cloud Computing,” IEEE Wireless Communications, vol. 20, no. 3, pp. 34–44, Jun. 2013.

[6] Y. Wang, X. Lin, and M. Pedram, “A Nested Two Stage Game Based Optimization Framework in Mobile Cloud Computing System,” in 7th International Symposium on Service Oriented System Engineering (SOSE). IEEE, March 2013, pp. 494–502.

[7] M. Ali, “Green Cloud on the Horizon,” in Cloud Computing, ser. Lecture Notes in Computer Science, M. Jaatun, G. Zhao, and C. Rong, Eds. Springer Berlin Heidelberg, 2009, vol. 5931, pp. 451–459.

[Online]. Available: http:// dx.doi.org/10.1007/978-3-642-10665-1 41

[8] G. H. Forman and J. Zahorjan, “The Challenges of Mobile Computing,” Computer, vol. 27, no. 4, pp. 38–47, 1994.

[9] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, “Saving Portable Computer Battery Power through Remote Process Execution,” Journal of ACM SIGMOBILE on Mobile Computing and Communications Review, vol. 2, no. 1, pp. 19–26, 1998.

[10] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic Resource Allocation and Parallel Execution in the Cloud fo.