ISSN 2395-1621

A Survey Paper On An Improved Scheduling Algorithm for Task Offloading On Cloud

#1Akash Kandarkar, #2Aditya Tornekar, #3Rahul Sawkar, #4 Madhurika Joshi

1kandarkar.akash@gmail.com
2adityatornekar@gmail.com
3rahulsawkar29@gmail.com
4madhu2264@gmail.com

#1234 Department of Computer Engineering, JSPM's ICOER

ABSTRACT

In the recent years, the usefulness of mobile devices has increased tremendously and with the limited energy capabilities of these devices this demand is not fulfilled. To suffice this demands we propose a concept of task offloading to cloud for reducing the energy consumption of device and to enhance the computation capabilities of devices. This paper's central idea revolves around the concept of task offloading onto the cloud platform from mobile devices and its factors which are useful in reduction of energy consumption. It also takes into consideration the scheduling and organization of tasks to be migrated on to the cloud so as to use the cloud resources for efficient computation of tasks in a time constrained environment. In the proposed system we classify the task into three categories (I) Task to be offloaded by default (II) Heavy resource utilizing tasks (III) Task which can be processed on the device itself.

Keywords—**Task Offloading, Virtual Machine(VM), Scheduling Algorithms, Cloud Computing.**

ARTICLE INFO

Article History

Received:5th October 2015

Received in revised form :

8th October 2015

Accepted:9th October, 2015

Published online :

13th October 2015

I. INTRODUCTION

Now-a-days smartphones have become our basic needs, but these smartphones have some limitations such as the battery life, the processing power of CPU, storage limitations etc. Over the last few years, rapid progresses in semiconductor technologies have reduced some of those constraints. While the smartphones are becoming increasingly popular because of their capabilities and functionalities, energy consumption of smartphones has also increased simultaneously. So, there is a need of framework to reduce the energy consumption. Cloud computing as we know is growing immensely, we can make use of this cloud computing technology to overcome the limitations of our smartphones such as storage, processing power of CPU and the battery life of our smartphones. By making use of task migration (offloading) we can migrate the processes from our smartphones to the cloud. Whenever a cloud provider receives a number of tasks from its user(s). This paper gives a brief about how these processes are offloaded to the cloud and what computations takes place on the cloud so as to give the output back to the smartphones.

Cloud is an architecture with gigantic amount of computational resources as well as

storage resources. Its main focus is On-demand resource availability. In cloud there are hundreds of servers with data centres. All the data stored can be accessible anytime as and when needed. Different types of clouds are available such as Public, Private, Hybrid and Community. These clouds are useful in their own ways and each cloud has its own specifications and limitations. Every cloud provides different services such as IaaS, PaaS and SaaS. The IaaS is infrastructure as a service (Amazon, EC2, Microsoft Azure), the PaaS is Platform as a service (Google App Engine), the SaaS is Software as a Service (Yahoo Mail, Google Apps, Microsoft Office Live)[6]. The main advantage of cloud is that we need to pay only for those services which we need.

Mobile Cloud Computing (MCC) is where all the data from the phone is sent to the cloud so as to overcome the limitations of phones. In smartphones many applications are running in the background, these application consume a lot of energy. This is the main reason for battery consumption. One way to overcome this is to migrate the tasks to cloud where the processing of tasks will take place instead of processing it on the phone. This concept comes under Cloud Migration. This paper serves as a collaborative view of the various algorithmic mechanisms and the related infrastructures, and identifies existing obstacles and scope for research. This survey paper is sectionalized as follows: Section 2 describes a brief history of Improvement in technology. Section 3 explains two objectives for offloading: reduction in execution time and low energy consumption, and explains infrastructures and softwares developed to address the hindrances of task offloading. Section 4 describes the research and the conclusion of this paper. As evident throughout this paper, many insights and research have been conducted on task offloading and a comprehensive survey of all related work is not feasible. Hence, this paper does not destine to give sufficient knowledge about topic. The references selected are based on knowledge of the topics, also providing a chronological order of this paper. Due to limitations of the length of the paper, this paper does not include some of the related papers.

II. IMPROVEMENT IN TECHNOLOGY

In recent years, the smartphone become one of the most essential devices on the hand of many people. The increment in the Internet, wireless communication, and the semiconductor technologies contribute to the popularity of the smartphones .The internal services and processes in smartphones are very complicated and difficult to understand .This rapid evolution of smartphones shows that the development of battery is not matching with development of hardware and software of smartphone which is developing magnanimously. Which divert the focus on Scheduling of process to utilize the power in proper manner so as to get long battery life.

Various Scheduling algorithms are used to reduce the overload from the processors of smartphones. In paper [1] a scheduler called application-aware scheduler (AGS) is developed. AGS avoid frequent replacement of cache to increase reduce system response time. Among the applications the execution is done alternately so as to achieve parallel processing, each process possesses thread. AGS manages the relation of threads belongs to a process to improve system performance. This paper [1] shows that AGS scheduler improves the system response time and looks more stable over old CFS scheduler. AGS [1] reduce response time up to 5.5%.

Coming over again to the power hungry devices i.e. smartphones, its need to satisfy application requirements regarding execution time. In paper [2] energy consumption is reduced used Energy-aware Dynamic Task Scheduling (EDTS), which uses the results of scheduling algorithm and sharply reduces energy consumption. This work in [2] results in high energy efficient working of system.

Further development in scheduling can be seen in [3] for astonishing work in scheduling algorithm. BPM-DFS [3] abbreviated as Bounded-Power Multicore Dynamic Frequency Scaling Scheduling Algorithm look after core mechanism, optimising working frequency and adjust core task execution and implies theoretical workload to practical workload for Predictive Power Model to estimate power consumption. Results from paper [3] shows power consumption difference between two built in power managers i.e. Linux and android. BPM-DFS [3] shows that it can save up to 25% of power consumption than Linux power manager.

Although many algorithm are invented for scheduling of tasks and services of smartphones. When it comes to networking it consumes most of the energy and reduces battery life. This can be

studied in paper [4] proposing models for estimating energy consumption. The details of network from lower networking layer to higher networking layer are considered. However, mobile devices are unable to accommodate most of the computing demand as long as they suffer the limited energy supply caused by the capacity of their small battery to store only a relatively small amount of energy. The literature describes several specialist techniques proposed in academia and industry that save the mobile device energy and solve this problem to some extent but not satisfactorily.

There is need of high performance which will perform tasks with no time latency and high performance level. This can only be achieved by offloading task to machine which is capable of doing such task. Cloud computing provides Platform as a Service, Software as a Service and Infrastructure as a Service. Their comparative study shows that there are different leading companies providing long list services. Offloading task to cloud i.e. Cloud Migration is pretty complicated task.

It is slightly different from the client-server architecture, here a thin client migrates computation to a server i.e. so called another Virtual Machine[10] on cloud having same configuration as smartphone Computation offloading is differs from the offloading model used in multiprocessor systems and distributed systems, where a process may be migrated for load balancing.VM work differently, main moto of VM management techniques is to properly managed VM such that resources are efficiently utilized. Paper [10] present VM management techniques for efficient utilization of resources such that it leads to reduce energy consumption and number of VM migration in virtualized data centres. There are number of research which have been carried out in this subject, some are practical based and some are based on simulation. By analysis of related work, we find that it is critical and essential to handle important things: first, how to allocate VM on host such that it is not over- loaded; second, which VM should be selected for migration from busy hosts. And last where to place (Reallocate) VM which is selected for migration.

The related work for the VM management is proposed in paper[9] the given algorithm find out the overloading host in virtual cloud and then decide which kind of methods and protocol we should follow so as to place the selected VMs. This algorithm procedure will lead us to proper management of VM such that energy consumption used by server centres as well as migration of VM is reduced.

Cloud Migration refers to a series of tasks performed to migrate an application into the cloud environment [6]. Not all tasks or applications are to be migrated to the cloud only those who consume more energy and require more computational load are migrated to the cloud. There are several different types of migration routes through which applications can be adapted to the cloud environment. Type I: replace, Type II: partially migrate, Type III: migrate the whole software stack, and Type IV: cloudify [6, 7, 8, 9].There are certain factors on which cloud migration depends [6], one of them is capability of cloud provider. There is a Cloud Client and a Cloud Provider co-operation between them is very important. While migrating tasks if the cloud client faces any problem (a problem which is not visible to the client, but is visible to the provider) then in that case the Cloud Provider helps the Cloud Client in solving those issues.

III. OFFLOADING DETERMINATION

Task offloading from mobile devices to cloud computing is a promising technique for tackling the problem especially with the emergence of high-speed wireless networks and the ubiquitous resources from the cloud computing. Paper [4] proposed an offloading framework to make task offloading possible to save energy for mobile devices. It achieved a great deal of progress toward developing a realistic offloading framework. First, that paper [4] examined the feasibility of exploiting the offloading technique to save mobile device energy using the cloud as the place to execute the task instead of executing it on the mobile device. Paper [4] study reveals that the offloading does not always save energy; in cases where the energy for the computation is less than the energy for communication no energy is saved. Therefore, the need for the offloading decision is vital to make the offloading beneficial. Second, paper [4] developed mathematical models for the energy consumption of a mobile device and its applications. We also studied paper related to models used for energy consumption estimation [11] which gives idea about every task which is to be offloaded, and also gives précised selection method for WLAN about which task is to be

offloaded and which to be not offloaded [11].Further models from [4] then used to develop mathematical models that estimate the energy consumption on the networking and the computing activities at the application level. Paper [4] modelled the energy consumption of the networking activity for the Transmission Control Protocol (TCP) over Wireless Local Area Network (WLAN).

Proposed framework from paper [13] supports task offloading of multiple services ongoing on android. If there are many services in one application running on system and all of those services can be offloaded to cloud, respective resource monitor indigently supports this situation and can make the respective allocation of task for execution. Enabling parallelization of the offloaded services on cloud is next motto. Furthermore, we can extend the current middleware framework so that it supports dividing arbitrary applications. It is difficult to estimate the characteristics of an Process or an application depending on different input/output parameters, which is exact connection between the input of the invoked task and the execution time took by it .The relationship between execution time and input parameters are characterised by running the target application several times on mobile system and adapt the task offloading algorithm accordingly.

While decision of offloading or deferring the processing of each task in a set of parallel tasks. There is précised formulation and solved Markov Decision Process (MDP) model for the mobile user to obtain an optimal offloading policy while minimizing the offloading and processing cost stated in paper [12].

Furthermore, paper [4] modelled the energy consumption of the computing activity for the mobile multi-core Central Processing Unit (CPU) and storage unit. It is identified and classified that the system parameters affecting the offloading decision and built our offloading framework based on them. In addition, paper [4] implemented and gave validation of the proposed framework by experimenting it on a real mobile device, VM cloud, and application. The experimental results shows that task offloading is more effective for smartphones given that in some cases it saves more than 70% of the energy required to compute a task. Additionally, proposed energy models accurately estimate the energy consumption for the networking and computing activities [4, 11]. This accuracy allows the offloading framework to make the correct decision as to whether or not offloading a given task saves energy. Framework described in [4, 11] shows it is built to be applicable to modern mobile devices and expandable by considering all system parameters that have impact on the offloading decision. In fact, the experimental validation proves that our framework is practical to real life scenarios. This framework gives researchers in the field useful tools to design energy efficient offloading systems for the coming years when the offloading will be common.

IV. CONCLUSION

This paper surveys and assorts a bombastic field of research related with task offloading for mobile systems. We analyse how facilitators like mobile agent, mobile modified framework and virtualization make task offloading feasible. We appraise different types of algorithms used to classify tasks and schedule them and offload tasks and related programs in order to improve efficiency of mobile or increase battery life. We considered the types of applications that have been used to demonstrate task offloading in this survey. We conclude on some of the research areas related to task offloading, and describe some models, frameworks and solutions that address these task offloading areas. In research we found there is a middleware framework called the Cloud Operating System (COS) which work as a server side technology for smartphones, which are client for that process. This back-end technology implements the notion of virtual machine (VM) flexibility to let cloud computing applications to scale up and down more effectively. In future the demand from the smartphones lead up to high performance real time applications like face detection for video. Executing task like this on smartphone will be more complex so, the offloading of task can be stretched up to task like running application like Computational Flow Dynamics (CFD).there is need of further development in utilization of GPUs by configuring high performance graphics cards. Sectionalisation of application on run time can be some with the help of actor oriented programming language like SALSA. Finally we describe why task offloading will become magnanimously important for resource constrained and connectivity devices in the future.

REFRENCES

[1]Z. Zhu, B. Sun, X. Li, X. Zhou, "Application-aware Group Scheduler for Android".

[2]M. Qiu, Z. Chen, L. T. Yang, X. Qin, B. Wang, "Towards Power-Efficient Smartphones by Energy-Aware Dynamic Task Scheduling".

[3]S. C. S. Chen, S. Weng, C. Li, "Design a Low-Power Scheduling Mechanism for a Multicore Android System".

[4]M. Altamimi, "A Task Offloading Framework for Energy Saving on Mobile Devices using Cloud Computing".

[5]S. E. Chang, K. Chiu, "Cloud Migration: Planning Guidelines and Execution Framework" 978-1-4799-8993-5/15/$31.00 ©2015 IEEE.

[6] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, "How to Adapt Applications for the Cloud Environment," Computing, vol. 95, no. 6, pp.493–535, Dec. 2012.

[7] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," IEEE Trans. Cloud Compute., vol. 1, no. 2, pp.142– 157, 2013.

[8] T. Binz, F. Leymann, and D. Schumm, "C Motion: A Framework for Migration of Applications into and between Clouds," 2011 IEEE Int. Conf. Serv. Compute. Appl. (SOCA), pp.1–4, 2011.

[9]S. Vahora, R. Patel, H. Patel, S. Patel, "Efficient Virtual Machine Management Based on Dynamic Workload in Cloud Computing Environment".

[10] M. Altamimi, A. Abdrabou, K. Naik, A. Nayak, "Energy Cost Models of Smartphones for Task Offloading to the Cloud".

[11]L. Gkatzikis, I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems".

[12]T. T. Hum, C. K. Tham , D. Niyato, "To Offload or to Wait: An Opportunistic Offloading Algorithm for Parallel Tasks in a Mobile Cloud" 978-1-4799-4093-6/14 © 2014 IEEE DOI 10.1109/CloudCom.2014.33.

[13]D. Kovachev, R. Klamma, "Framework for Computation Offloading in Mobile Cloud Computing".

[14]Y. Z., H. Liu , L. Jiao, X. Fu, "To offload or not to offload: an efficient code partition algorithm for mobile cloud computing"

