Proceedings of the 23rd International Conference on Automation & Computing, University of Huddersfield, Huddersfield, UK, 7-8 September 2017

Effective Task Migration to Reduce Execution Time in Mobile Cloud Computing

Sajeeb Saha[1] and Mohammad S. Hasan[2]

[1]Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh

[2]School of Computing and Digital Technologies, Staffordshire University, Stoke-on-Trent, UK

***Abstract*— With the advancements of mobile technologies, different compute-intensive tasks are emerging rapidly. However, due to resource constraints, these tasks are facing challenges to execute on mobile devices. As a solution to this problem, cloud migration has been introduced to execute a task on the cloud and then to return the results to the user mobile device. In this paper, a cloud migration decision making algorithm for compute-intensive tasks has been proposed to determine the feasibility of execution on a cloud server instead of a mobile device. Furthermore, the performances between mobile and cloud executions have been investigated which shows that the task completion time can be minimised by 6-8 times when a cloud server is utilised.**

Keywords- compute-intensive; task migration; mobile cloud; decision algorithm; cloud computing;

I. INTRODUCTION

The advancement of mobile computing technologies has paced the usage of mobile devices (smartphones, tablets, wearable devices etc. [1]) that are equipped with various sensors to support versatile compute-intensive tasks in the field of image processing, multimedia, wearable computing, augmented reality, mobile healthcare etc. A recent study shows the number of mobile application downloads all over the world in 2016 was 149.3 billion which is expected to be 352.9 billion in 2021. These applications produced a revenue of $88.3 billion that is estimated to be around $188.9 billion by the end of 2020 [2]. Though mobile devices are equipped with powerful hardware now-a-days, they often cannot cope with some recent applications requiring large amount of processing power and memory.

Fortunately, mobile cloud computing has been introduced to alleviate these constraints and to support the ever increasing computation and storage demand for mobile devices [3]. Mobile cloud computing allows to migrate compute-intensive applications or tasks to a remote cloud server which offers huge amount of resources for computation and storage. Generally, compute-intensive tasks perform a large amount of complex computation on small amount of data on a mobile device and thus consume larger amount of energy [4]. Furthermore, execution of many recent tasks is not feasible on mobile devices as they cannot meet the minimum time requirement. The cloud based execution is beneficial for these tasks since execution time can be reduced vividly compared to the amount of data transfer for the instance. Therefore, migration algorithms for cloud based execution are immerging for such tasks. However, a major challenge in task migration is the accurate prediction of migration feasibility as the task migration incurs additional overhead e.g. time for network communication and data transfer. The main contributions of this paper are summarised as follows.

- A decision making algorithm has been proposed and implemented to determine the feasibility to migrate a task from a mobile device to a cloud server to minimise execution time.
- An experimental testbed with Amazon EC2 cloud server [5] to conduct the experiments.

The rest of the paper is organised as follows. Section II contains a review of related cloud based task migration mechanisms. A detailed description of the proposed task migration decision algorithm is presented in Section III. Section IV presents the implementation details and evaluates the performance of the algorithm from the results obtained. Finally, some conclusions are drawn in Section V.

II. RELATED WORKS

A number of works already have done on task migration technique to reduce execution time of a task. In [6] Chun et al. proposed a flexible task partitioning system to migrate and execute part of a task on a remote server. It used static analysis and dynamic profiling for partitioning tasks to reduce the execution time and energy use. A framework named ThinkAir was proposed by Kosta et al. in [7] for migrating mobile tasks to cloud using a method-level migration and smartphone virtualisation technique. The work employed elastic and scalable Virtual Machine (VM) images to execute tasks in parallel and produced increased overall throughput. In [8], Barbera et al. evaluated performance of an architecture which contained two clone copies of a task namely off-clone and back-clone stored on a cloud server and were used for migration and backup, respectively. A real testbed with 11 smartphones and an Amazon EC2 public cloud instance were used to measure the performance of a system in terms of bandwidth and energy consumption. A summary of the promising computational migration technologies and architectures were summarised by Kumar et al. in [9] where authors focused on different algorithmic approaches to make migration decisions to save energy and to improve performance of the mobile system. Furthermore, recent mobile cloud computing architecture and task models were highlighted by Khan et al. with their pros and cons in [10]. In [11], Cheng et al. proposed a cloud migration architecture and an algorithm to migrate a portion of computation tasks from wearable devices to

local mobile devices or to remote cloud so that heavy computation tasks can be executed.

Task migration requires computation as well as a large amount of data transfer. In general, compute-intensive tasks require a small amount of data transfer compared to huge amount of complex processing. Moreover, the wireless communication technologies also have evolved to support several megabits for cellular networks to several Gigabits for Wi-Fi networks [12], [13], [14]. In a recent work [15] Khoda et al. showed that task migration from a local mobile device to a remote cloud can maximise energy saving while maintaining strict latency requirements of user tasks in 5G systems. Hence, this paper focuses on a migration decision making algorithm which executes a task on the cloud if it is beneficial compared to the execution on the mobile device.

III. THE PROPOSED TASK MIGRATION ALGORITHM

This section describes a mathematical formulation for compute-intensive task execution in cloud platform. The architecture to facilitate the migration from a mobile device to a cloud server is presented and an algorithm to decide the feasibility of a migrated execution is proposed.

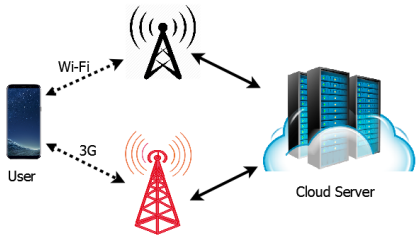

Figure 1: Task Migration Architecture.

A. Task Migration Architecture

Compute-intensive tasks require a large amount of CPU cycles which make them hard to execute on mobile devices. This paper considers a two-layer architecture with mobile devices. Tier 1 is responsible to initiate the migration to a remote server, if suitable and tier-two is equipped with higher processing capacity as depicted in Figure 1. Based on the Service Level Agreement (SLA) with mobile device, the cloud server provides a virtual machine instance with specified amount of resources to execute migrated tasks. Whenever a mobile device needs to execute a compute-intensive task, based on the size of the task, it estimates the execution time on the device and the completion time on the cloud that requires cloud execution time and communication delay. The system uses previous connection establishment history between the mobile device and the cloud server to estimate the communication delay. For communication and data transmission, both 3G and Wi-Fi technology has been considered. However, any other network communication technology is applicable. The mobile device migrates a task to the cloud if the estimated cloud completion time is smaller than the mobile device execution time. The cloud server executes the migrated task and sends results back to the mobile device if resources are available. Otherwise the cloud server returns a failure notification to the mobile device and the task is executed on the mobile device. The complete flow chart of the decision making and task execution is shown in Figure 2.

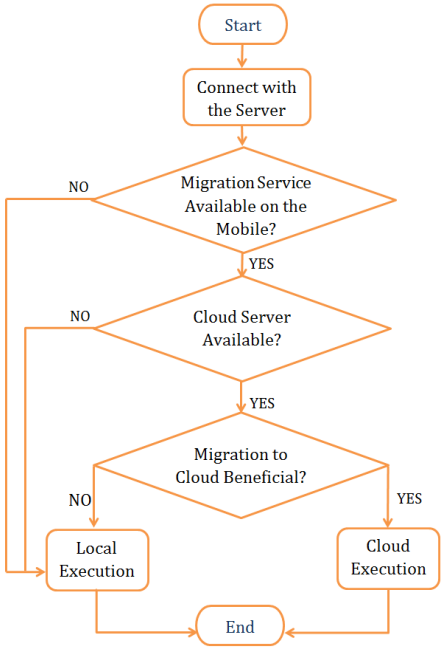

Figure 2: Work flow for the migration decision making process.

B. Migration Decision Formulation

To execute a task on the remote cloud server, at first the mobile device makes a migration request to the cloud server. The task size estimation considers the number of key instruction $\boldsymbol{N}$ of the task. Let $\boldsymbol{\beta_{mobile}}$ and $\boldsymbol{\beta_{cloud}}$ denote the execution time per key instruction on the mobile device and cloud server, respectively which are available to the mobile device before making the request. Therefore, the mobile device estimates the mobile device execution time ($\boldsymbol{T_{mobile}}$) and cloud computation time ($\boldsymbol{E_{cloud}}$) using equation (1) and (2).

$$T_{mobile} = N * \beta_{mobile} \quad ----------(1)$$

$$E_{cloud} = N * \beta_{cloud} \quad ----------(2)$$

To execute the task on the cloud server, a communication time ($\boldsymbol{C_{cloud}}$) is required that includes connection establishment time and input/output data transfer time between the mobile device and the cloud server. Hence, task completion time on the cloud ($\boldsymbol{T_{cloud}}$) is calculated using equation (3).

$$T_{cloud} = E_{cloud} + C_{cloud}; \quad ----------(3)$$

Based on the mobile execution time and the estimated cloud completion time, the mobile device decides that migration is favorable if $\boldsymbol{T_{cloud}}$ is less than $\boldsymbol{T_{mobile}}$ and the task is migrated to the cloud server. The sequence of steps for the whole decision making algorithm is summarised in **Algorithm 1**.

Algorithm 1*: *Algorithm for compute – intensive task migration decision

$INPUT: N, \beta_{mobile}, \beta_{cloud}, C_{cloud};$

$OUTPUT: result\ or\ failure\_notification;$

$$\boldsymbol{function\ cloud\_migration}()$$
{
$T_{mobile} = N * \beta_{mobile};$
$E_{cloud} = N * \beta_{cloud};$
$T_{cloud} = E_{cloud} + C_{cloud};$
$if(T_{cloud} < T_{mobile})$
{
$if(VM \neq BUSY)$
{
$result = execute\_application();$
$return\ result;$
}
$else$
{
$return\ resource\_not\_available;$
}
}
$else$
{
$return\ migration\_failure;$
}
}

IV. Experimentation Setup and Results

This section provides a detailed explanation of the experimental setup and the results.

A. Experimental Testbed

To evaluate the execution time improvement on the cloud, an experimental testbed has been developed based on the proposed algorithm. The Bubble sort algorithm has been used to emulate a compute-intensive task on a mobile device. A number of unsorted array elements are transferred from the mobile device to the cloud server and the sorted array is sent back to the mobile device using the communication network. To provide the cloud server functionality, an Amazon Elastic Compute Cloud (EC2) instance has been utilised. Wi-Fi and 3G communication technologies have been used to migrate compute-intensive tasks to cloud. When tasks are executed on mobile device, only mobile execution time is treated as completion time. On the other hand, cloud completion time includes computation and communication times. To observe the effectiveness of the migration decision making algorithm, the emulation system uses two set of compute-intensive tasks based on small array sizes and large array size. Small array sizes range from 1000 to 6000 (4 KB to 24 KB) and large array sizes range from 10000 to 60000 (40 KB to 240 KB). The experimental results for small array sizes help to identify the cutoff point between mobile execution and cloud migration, whereas large array sizes are used to identify the amount of potential gain in terms of completion time through cloud migration. All the experiments have been conducted for 20 times and the obtained results are averaged. Confidence interval has been introduced in calculation to measure the tendency and timing variation in the task execution. The details of the implementation settings and parameters are given in Table 1.

TABLE 1: DEVICE SETTINGS

Device	Model	OS Version	RAM	CPU
Smart phone	Sony LT18i	Android 4.0.4	512 MB	1.4 GHz Scorpion Number of Core:1
Cloud Server	Amazon EC2 Instance	Microsoft Windows 2012 R2 Standard Edition (64-bit)	1GB	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz Number of Core:8

B. Impact on the Small Array Sizes

First, the experiments are conducted between cloud completion with Wi-Fi connection and mobile execution to evaluate the performance of the proposed algorithm in section III.B. and the results are shown in Figure 3. It can be noted that the algorithm executes the task on the mobile device for array sizes 1000 and 2000. However, for array size 3000 or larger, it migrates the task to the cloud server.

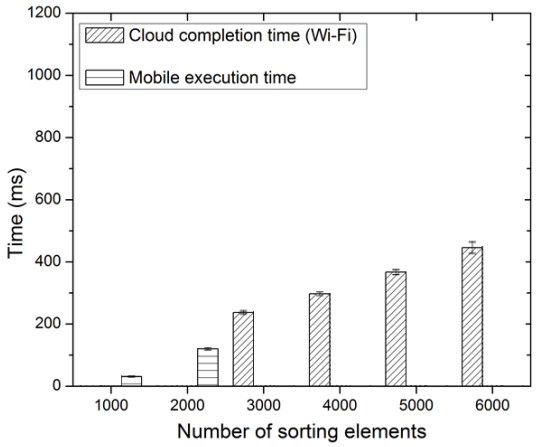

Figure 3: The performance of the proposed algorithm between cloud with Wi-Fi and mobile execution.

Figure 4 shows the decision making performance of the proposed algorithm between cloud with 3G and mobile execution. It can be noted that the algorithm executes the sorting task on the mobile up to array size 4000 and then migrates the task to the cloud for larger array sizes.

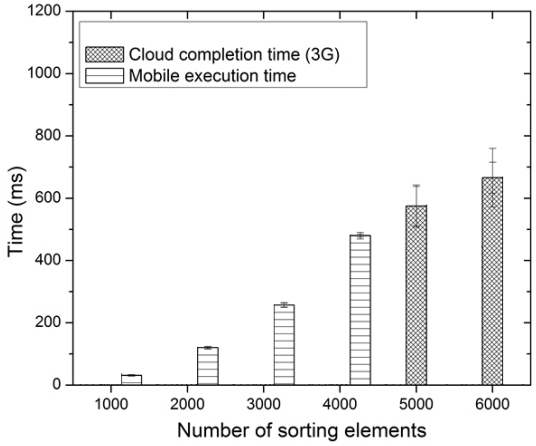

Figure 4: The performance of the proposed algorithm between cloud with 3G and mobile execution.

To understand the decision making performance of the proposed algorithm, all the required times for cloud with

Wi-Fi and 3G and mobile execution are plotted in Figure 5. It shows that the task completion time increases linearly as the array size increases. In case of the array sizes up to 2000, mobile execution outperforms cloud based execution irrespective of communication technologies. It can be noted that the preference of mobile execution can be extended to array sizes up to 4000 for 3G networks. However, for array sizes of 5000 and onwards, the system always produces result in favor of the cloud migration.

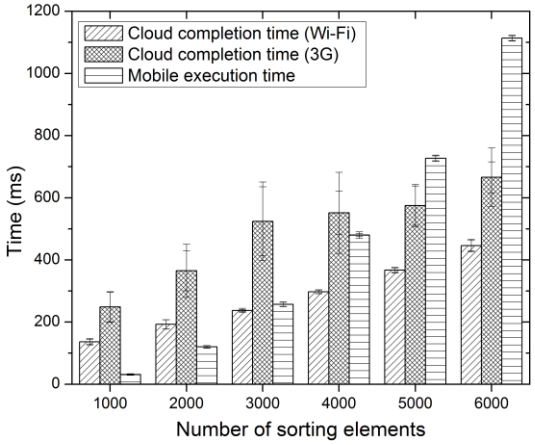

Figure 5: Task completion time for small array sizes.

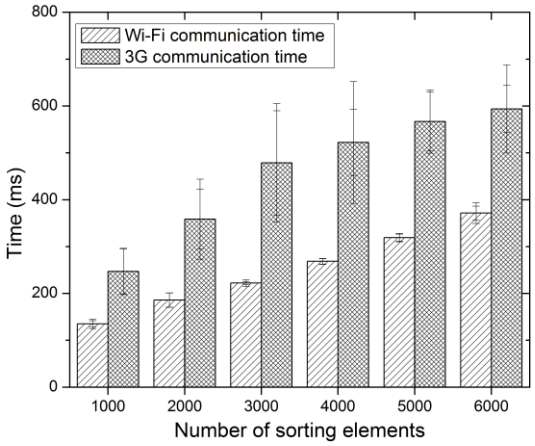

Figure 6: Communication time required for small array sizes

The impact of communication technologies (3G and Wi-Fi) during migration is shown in Figure 6. The result shows linear increase in communication time with the increase of the array size for both Wi-Fi and 3G. However, Wi-Fi is always better than 3G which is also expected theoretically. It has been observed that with the increasing array size, communication time increases due to higher amount of data transfer. Moreover, the 3G communication latency shows higher variance than Wi-Fi.

Figure 7 shows that the cloud computation time is several times lower than the mobile execution due to more powerful computation resources on the cloud server. Although cloud execution provides much better result, mobile device execution is still beneficial before cutoff point i.e. array size of 2000 to 4000 as shown in Figure 3. This is due to large amount of communication overhead which suppresses the gain in migrated computation on the cloud server. However, after the cutoff point, cloud migration is always beneficial as cloud completion time alleviate the communication overhead with faster execution capability compare to mobile device execution.

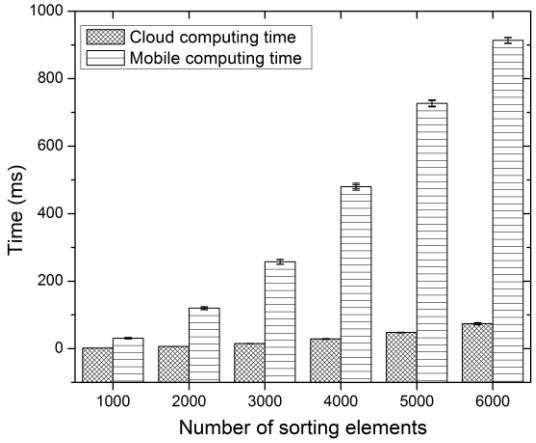

Figure 7: Computation time required for small array sizes

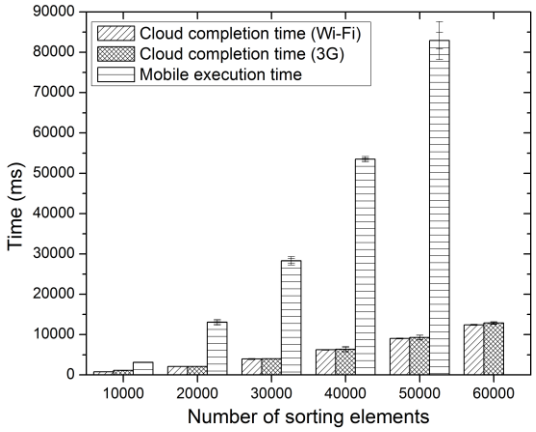

Figure 8: Task completion time for large array sizes

C. *Impact on the Large Array Sizes*

Figure 8 shows that the completion time with various array sizes. In case of cloud, the completion time increases slightly with the array size whereas completion time increases rapidly for the mobile execution. It has been noted that that mobile device fails to execute the task of Bubble sort for array size 60000 or larger and hence does not show any output in the graph. Cloud execution exhibits very similar results for Wi-Fi and 3G.

To provide a deeper insight on how cloud computing outperforms the mobile device, communication and computation time are extracted from the results. The communication graph, Figure 9, shows that as the array size increases, communication time grows as expected. It has been noted that the majority portion of the communication time is spent on data transfer rather than connection establishment.

In case of cloud execution, due to the availability of more powerful resources, computation time can be reduced by 6-8 times compared to mobile based execution which is shown in Figure 10. Hence, with the increasing array size, the communication time does not have significant impact on the overall performance gain for

migration and remains several times higher than the mobile device execution.

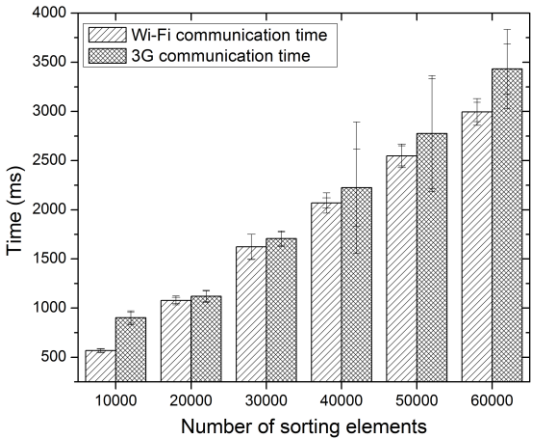

Figure 9: Communication time required for large array sizes

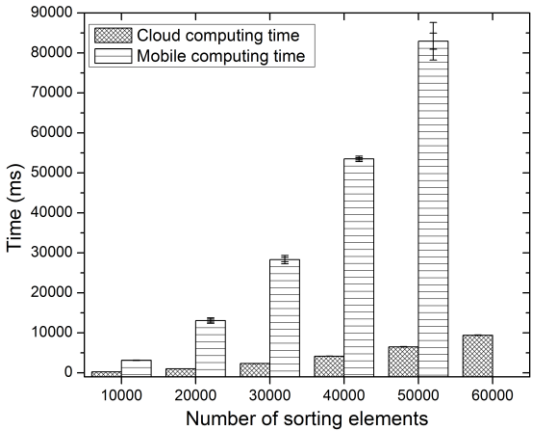

Figure 10: Computation time required for large array sizes

V. CONCLUSION

In this paper, a decision making algorithm to determine the feasibility to migrate a task to a cloud server for compute-intensive tasks has been proposed and implemented. The results show that for tasks with high computation and small amount of data transfer the cloud outperforms mobile device regardless of network communication technologies. As the future works, an extension of the algorithm to work with multiple virtual machines and parallel execution of task modules to further reduce the execution time of different tasks, various types of tasks are being considered by the authors.

VI. ACKNOWLEDGEMENT

The authors would like to acknowledge an Erasmus+ International Credit Mobility (ICM) for Bangladesh fund awarded to Staffordshire University, UK in 2016.

REFERENCES

[1] Mark Billinghurst and Thad Starner. Wearable devices: new ways to manage information. Computer, 32(1):57–64, 1999.

[2] Global mobile app usage statistics. https://www.statista.com/topics/1002/mobile-app-usage/, Access Date: 10 May 2017.

[3] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing, 13(18):1587–1611, 2013.

[4] Junjie Peng, Yongchuan Dai, Yi Rao, Xiaofei Zhi, and Meikang Qiu. Modeling for cpu-intensive applications in cloud computing. In High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on, pages 20–25. IEEE, 2015.

[5] EC Amazon. Amazon elastic compute cloud (amazon ec2). Amazon Elastic Compute Cloud (Amazon EC2), 2010.

[6] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic Execution Between Mobile Device and Cloud," Proc. sixth Conf. Comput. Syst., pp. 301–314, 2011.

[7] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In Infocom, 2012 Proceedings IEEE, pages 945–953. IEEE, 2012.

[8] Marco V Barbera, Sokol Kosta, Alessandro Mei, and Julinda Stefa. To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In INFOCOM, 2013 Proceedings IEEE, pages 1285–1293. IEEE, 2013.

[9] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, and Bharat Bhargava. A survey of computation offloading for mobile systems. Mobile Networks and Applications, 18(1):129–140, 2013.

[10] Mazliza Othman, Sajjad Ahmad Madani, Samee Ullah Khan, et al. A survey of mobile cloud computing application models. IEEE Communications Surveys & Tutorials, 16(1):393–413, 2014.

[11] Zixue Cheng, Peng Li, Junbo Wang, and Song Guo. Just-in-time code offloading for wearable computing. IEEE Transactions on Emerging Topics in Computing, 3(1):74–83, 2015.

[12] Puneet kumar and Er ManwinderSingh. 5g technology of mobile communication. International Journal of Electronics and Computer Science Engineering, 2(4):1265– 1275, 2012.

[13] Eldad Perahia, Carlos Cordeiro, Minyoung Park, and L Lily Yang. Ieee 802.11 ad: Defining the next generation multi-gbps wi-fi. In Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, pages 1–5. IEEE, 2010.

[14] Oscar Bejarano, Edward W Knightly, and Minyoung Park. Ieee 802.11 ac: from channelization to multi-user mimo. IEEE Communications Magazine, 51(10):84–90, 2013.

[15] Mahbub E Khoda, Md Abdur Razzaque, Ahmad Almogren, Mohammad Mehedi Hassan, Atif Alamri, and Abdulhameed Alelaiwi. Efficient computation offloading decision in mobile cloud computing over 5g network. Mobile Networks and Applications, 21(5):777–792, 2016.