IOP Conference Series: Materials Science and Engineering



# An enhanced technique for mobile cloudlet offloading with reduced computation using compression in the cloud

To cite this article: A C Moro and R K Nadesh 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042075

# An enhanced technique for mobile cloudlet offloading with reduced computation using compression in the cloud

**Moro A C and Nadesh R K**

School of Information Technology and Engineering, VIT University, Vellore-632014, Tamil Nadu, India.

Email: rknadesh@gmail.com

**Abstract** The cloud computing paradigm has transformed the way we do business in today's world. Services on cloud have come a long way since just providing basic storage or software on demand. One of the fastest growing factor in this is mobile cloud computing. With the option of offloading now available to mobile users, mobile users can offload entire applications onto cloudlets. With the problems regarding availability and limited- storage capacity of these mobile cloudlets, it becomes difficult to decide for the mobile user when to use his local memory or the cloudlets. Hence, we take a look at a fast algorithm that decides whether the mobile user should go for cloudlet or rely on local memory based on an offloading probability. We have partially implemented the algorithm which decides whether the task can be carried out locally or given to a cloudlet. But as it becomes a burden on the mobile devices to perform the complete computation, so we look to offload this on to a cloud in our paper. Also further we use a file compression technique before sending the file onto the cloud to further reduce the load.

## 1. Introduction

New technologies are being developed at a fast pace in today's world. With technologies such as Cloud-Computing, Big Data and many others, more development is streamlined into these areas. With greater adoption to these technologies, greater is the market value. One such technology is that of cloud-computing which promises to grow in the coming areas to a large extent. Through cloudcomputing storage, databases, networking, software and more are delivered over the Internet. As the cloud can and hybrid models, it can be built suited for any model requiring cloud services.

Apart from the SAS, PAS and IAAS services of the cloud, other obvious features include greater security and lesser load on your hard drive. Our work is basically related to offloading to cloudlets depending upon multiple factors which are elaborately stated in the forthcoming sections. Offloading is a broad term related to transferring certain computing and tasks to nearby cloudlets when the local system lacks the hardware capacity to do so. Broadly, this means almost every task can be offloaded to the cloud whether it is pushing the local system hardware limits or not. Hence, we are faced with certain restrictions when we try to offload.

However, it is not necessary that all tasks should be offloaded. There has to be a factor or a threshold to determine if the computation can be done locally or has to be offloaded. This threshold cannot depend upon a single ruling criteria, but will be based upon multiple criteria set through multiple paths. These factors include availability of the cloudlets in the vicinity of the user device, mobility of both the user and the cloudlets with respect to each other, admission condition of the cloudlets. After these we have other factors such as power consumption and cost. Then we have to look at other dynamic factors such as loss in connection while offloading or when the processed data is returned to the local device.

Even with such a high rate of failure, offloading has many advantages and so, it becomes critical to increase the chances of offloading and getting the result back successfully. For this we use Markov's Decision Process(MDP), Poisson's process and an optimal algorithm to determine a threshold value. This process is the most critical in this project. If any job has to be offloaded as opposed to being executed locally, then we take further take steps to decrease the load on the cloud as well by compressing the file before it is sent to the cloud. The chief compression technique used here is the. Zip type of file compression. This way, even large sized files and programs can be offloaded to the cloud without putting too much load on the cloud's resources.

## 2. Methodology

The proposed method involves many phases or stages. The first major phase involved is finding the threshold value required to determine if a particular job can be executed locally or has to be offloaded.

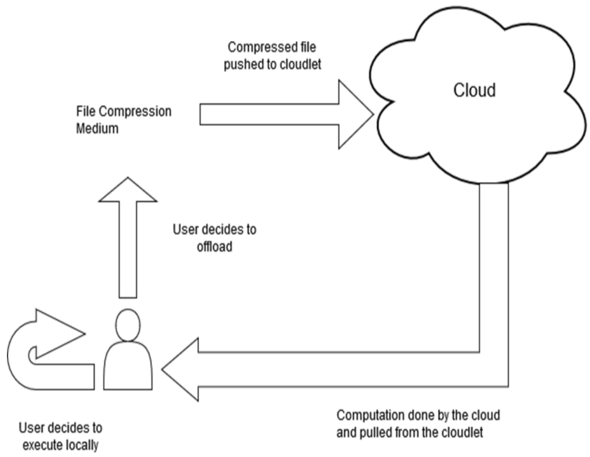


Figure 1. System Architecture

The system architecture follows the following steps:
I. User has mobile memory and data ready for offloading
II. Based on factors such as cost, energy estimations and availability of cloudlets for offloading, an agreement is reached whether to carry out the execution of the job on the device itself or on the cloud.
III. If the job is to be executed locally, then it is done on the device itself.
IV. If the job is to be offloaded, then availability of cloudlets is checked.
V. The file to be offloaded is compressed to a format like. Zip.
VI. The user logs in and the offloading process begins
VII. The computation is carried out by the cloud and the result is returned to the mobile.
VIII. Finally, the user logs out once the process is completed.

### 2.1. Calculating The Threshold

For calculating the system threshold, it is necessary to understand the system state which is given as:

$$S= (G, Q, N)$$

The above formula represents the synthesized state of a user, including application part of the process, the queue state, and also the strength in numbers of cloudlets. Based on this complete state, the user

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042075 doi:10.1088/1757-899X/263/4/042075

calculates both the immediate and the local costs. Based on these costs, the end user takes a calculated judgement A(S) i.e. decision in state S, whether to execute the job locally(A=0) or offload it(A=1).

### *2.2. Concept of Markov Decision Process and Poisson Point Process*

A Markov Decision Process(MDP) includes a set of possible states(S), a set of actions(A), a real valued reward function R (s, a) and a detailed explanation of the effect of every action in every state. In this model, it is naturally assumed that an action's effects taken for that state rely solely on that particular state and not on the other previous states. A MDP model may be solved with either linear or dynamic mathematical programming. In our case we use Bellman equations for formulation.

The Poisson Point Process(PPP) is related to the Poisson Distribution process, which implies that the probability of a Poisson random variable N is equal to n given by:

$$P\ \{N = n\} = (\Lambda^n/n!\ )\ e^{-\Lambda}$$

Where n! denotes n factorial and $\Lambda$ is the single Poisson parameter that is used to define the Poisson distribution. The other key property is that for a collection of disjoint (or non-overlapping) bounded sub-regions of the underlying space, the number of points in each bounded sub-region will be completely independent of all the others.

### *2.3. Intermittent Connection Issues:*

Connection issues between the mobile device and the cloud may include the following:
I. Availability of cloudlets in the vicinity of the mobile user.
II. Admission policy of the cloudlet based on the resources on the cloudlet.
III. Relative mobility between mobile and cloudlet.
IV. Availability of good internet Wi-Fi/3G connection.
V. Time taken for offloading to complete.

### *2.4. ZIP Compression of files*

Once the user decides to offload to the cloudlet, we use file compression technique i.e. .zip to compress the file before it is offloaded onto the cloudlet. Like other compression archives, ZIP files are data containers, i.e. they store one or several files in the compressed form. In this ZIP format, it becomes very easy to keep related files together and makes transferring through methods such as E-mails, retrieving and storing information and software; quicker and a lot more economical.

Compressed files such as Zip files use several different compression methods. These include deflate, deflate64, bzip2, lzma, wavpack, and ppmd. In practice however, it will almost always use deflate exclusively in zip files, for compatibility. Deflate is also the compression method used in GZIP and by zlib, as well as by the PNG image format. Deflate is an LZ77(Lempel – Ziv) compressor.

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042075 doi:10.1088/1757-899X/263/4/042075

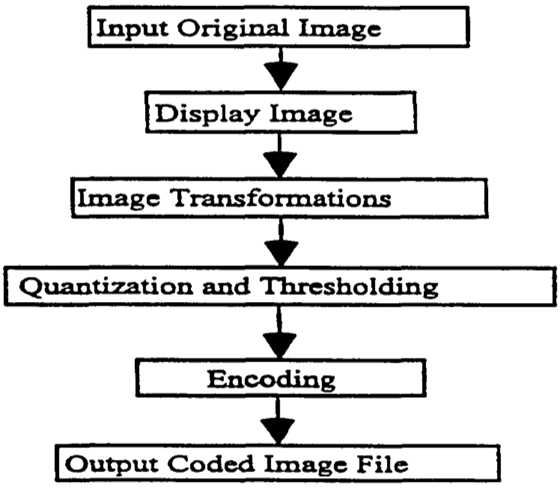


Figure 2 . Compression Flowchart

The tasks considered for this purpose include addition, subtraction, multiplication, face detection and feature extraction. These tasks have varying complexity and are thus, configured for the research. For optimal functioning, A = 0 for addition, subtraction and multiplication and A = 1 for face detection and feature extraction. Each of these tasks is assigned a jobId when it enters the queue for execution for current state S. Based on the jobId and its location in the queue, it will be given a priority for the sequence of tasks to execute.

The threshold factor is based on the following factors:

I. CPU power: Power consumed by the CPU of the local device.

II.CPU time and CPU Speed: Time required by the CPU to complete the tasks given in ms. Delay can be considered as a varying factor here. The speed of the CPU depends upon the bandwidth and the number of available processors.

III. Offloading energy: This describe the least quantity of power/energy required for the local user mobile to offload to nearby cloudlet. This further depends upon factors such as distance(D) between the device and the cloudlet, the size of the job to be offloaded, local Internet signal strength and acceptance policy of the crowd.

IV. Cloud Usage: If a cloudlet's resources are already full then, the request from the user will be rejected and user has to wait for a period of time before he can offload.

V. Bandwidth: Bandwidth size and how the bandwidth is allocated has a large effect in optimized offloading process.

We configure this using Eclipse Juno IDE to simulate in CloudSim environment.

## 3. Result and Discussion

The tests have been carried out given the two scenarios:

I. Offload everything to the cloud, or

II. Offload only those tasks that are too heavy on the local memory and the processors and execute the rest locally.

Given the first scenario, even the most basic tasks that is addition, subtraction and multiplication are being offlaoded to the cloud. Although this reduces the workload on the local device, it is not the most optimal way of working. From the list of tasks we have built to be simulated,

We have to first configure the mobile centers and the data centers with respect to the boundary regions.

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042075 doi:10.1088/1757-899X/263/4/042075

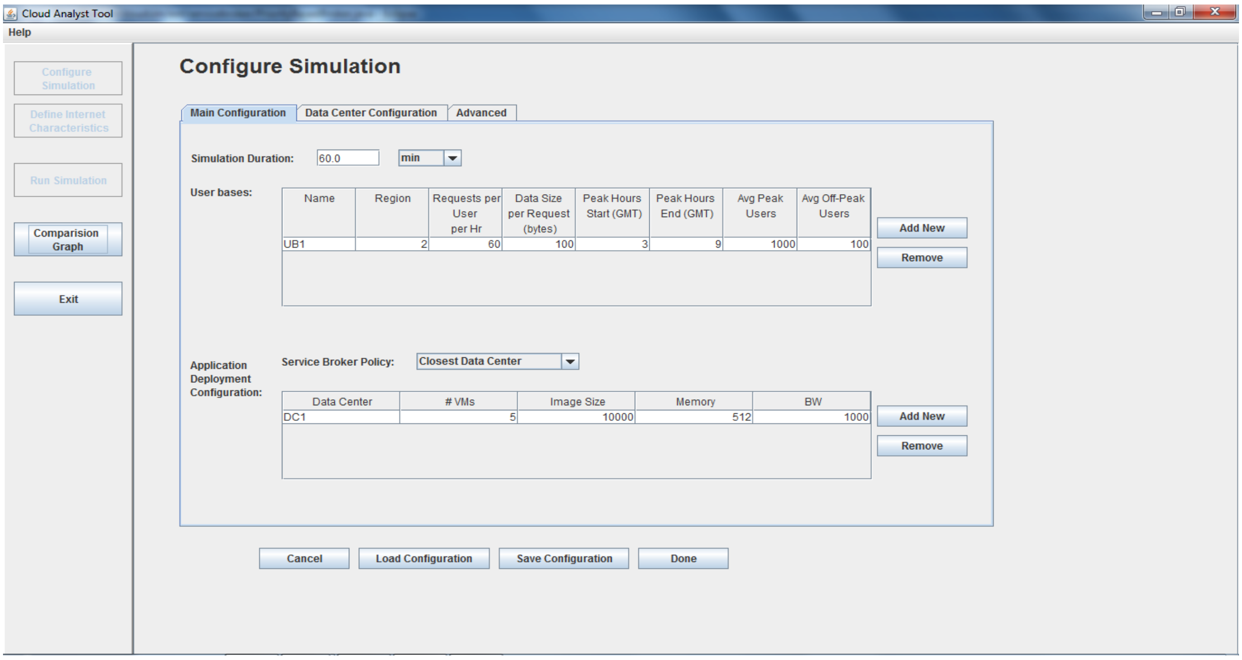


Figure 3 . Configuration Menu

Initially, we have a predefined set of configuration for a single region only. We can add or remove mobile centers and data centers for the simulation as required. The final configuration will look like Figure 3.

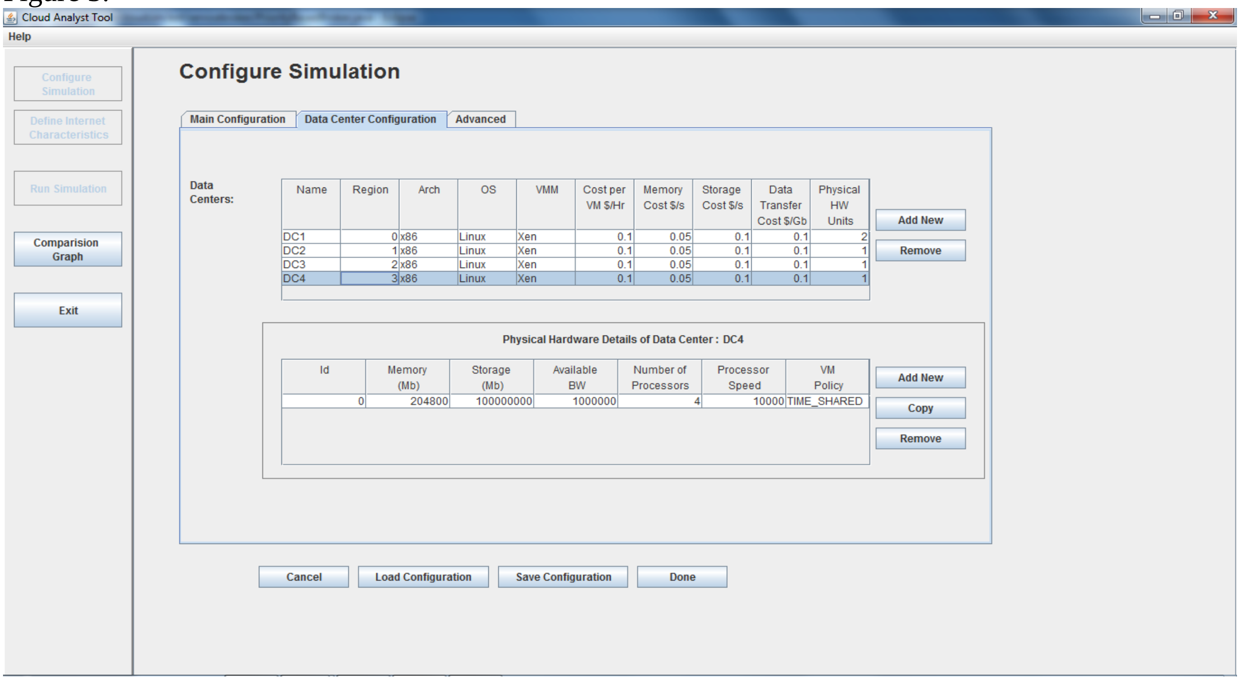


Figure 4 . Hardware Specifications

One important factor here is the service-broker policy. This is used to choose between the closest data center, optimal response time or dynamic reconfiguration. From the results, we can note down the overall response time which includes data center processing time and power consumption.

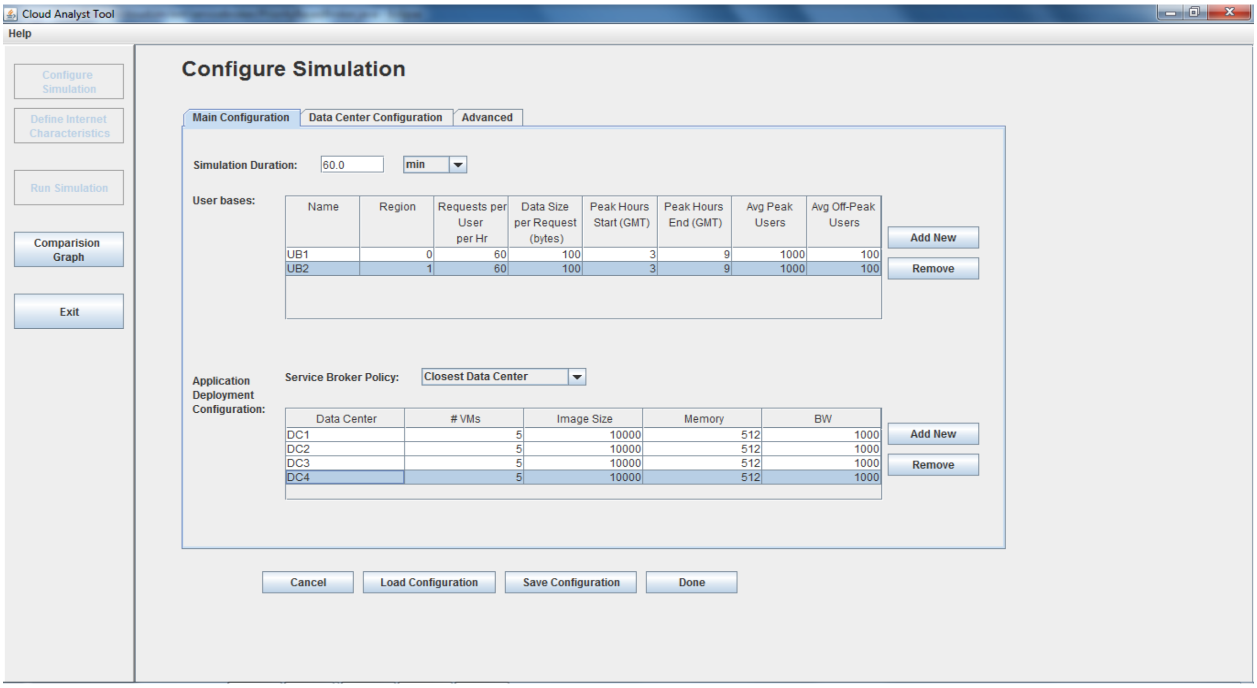


Figure 5 . DataCenter Configuration

Further segregation includes response time by region. If we compare the data center servicing time for requests, we can see how the requests are processed in lesser time periods in case of the optimal algorithm.

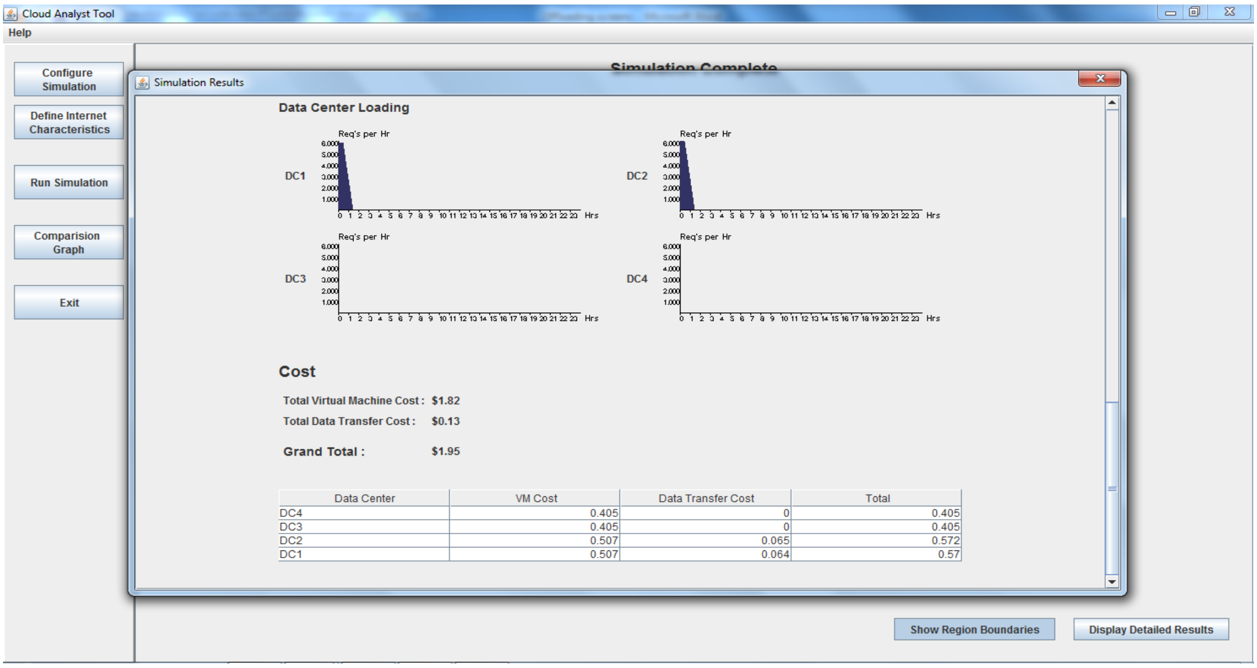


Figure 6: Cost Matrix

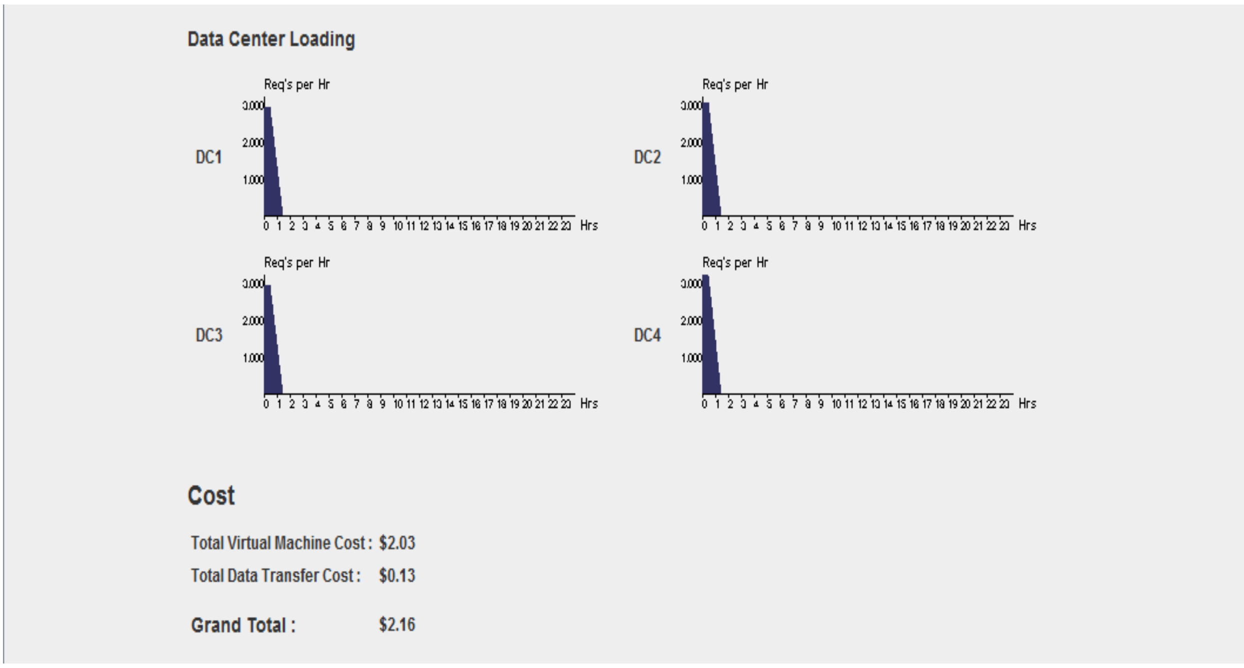


Figure 7 Data Center

If we do a final comparison of the cost in Figure 6 and Figure 7, we can see the difference in costs. This is in agreement with the optimal algorithm stated earlier. Also, further by compressing the files, the processing periods increase slightly but serves at a lower cost.

## 4. Conclusion

Further research can be done in this field which involves varying the compression technique. A lossless compression technique wherein the ratio of the data compressed to the original file size is required. Another area of investigation is the file type that is getting compressed. It is not necessary that all file types will give the same desired result. Lastly, the case of file compression in the cloud requires a separate work area in the cloud computing paradigm.

## References

[1] Zhou B, Dastjerdi A V, Calheiros R N and Buyya R 2015 A Context Sensitive Offloading Scheme for Mobile Cloud Computing Service *IEEE 8th International Conference on Cloud Computing* 869-876

[2] Melendez S, McGarry M P, Teller P J and Bruno D 2015 Communication Patterns of Cloud Computing *Globecom Workshops (GC Wkshps) IEEE DOI: 10.1109/GLOCOMW.2015.7414096*

[3] Yrjo R and Rushil D 2011 Cloud Computing in Mobile Networks – Case MVNO. 15thInternational Conference on Intelligence in Next Generation Networks DOI: 10.1109/ICIN.2011.6081085

*[4]* Zhao Y, Zhou S, Zhao T and Niu Z 2015 Energy-Efficient Task Offloading for Multiuser Mobile Cloud Computing *IEEE/CIC ICCC 2015 Symposium on Selected Topics in Communications DOI: 10.1109/ICCChina.2015.7448613*

*[5]* Wu H, Wang Q and Wolter K 2012 Methods of Cloud Path Selection for offloading in Mobile Cloud Computing Systems *IEEE 4th International Conference on Cloud Computing Technology and Science DOI: 10.1109/CloudCom.2012.6427587*

*[6]* Sharma R, Kumar S and Trivedi M C 2013 Mobile Cloud Computing: A Needed Shift from

Cloud to Mobile Cloud *5th International Conference on Computational Intelligence and Communication Networks DOI: 10.1109/CICN.2013.116*

*[7]* Routaib H, Elmachkour M, Eikoutbi M, Sabir M E and Badidi E 2014 Modeling and Evaluating a Cloudlet-based Architecture for Mobile Cloud Computing *Intelligent Systems: Theories and Applications (SITA-14) DOI: 10.1109/SITA.2014.6847290*

[8] Zhang Y, Niyato D and Wang P 2015 Offloading in Mobile Cloudlet Systems with Intermittent Connectivity IEEE Transactions On Mobile Computing **14**(12) 2516-2529

*[9]* Jararweh Y, Jordan F, Tawalbeh L and Dosari F 2013 Resource Efficient Mobile Computing using Cloudlet Infrastructure *9th International Conference on Mobile Ad-hoc and Sensor Networks*

*[10]* Barbera M V, Kosta S, Mei A and Stefa J 2013 To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing *Proceedings IEEE INFOCOM DOI: 10.1109/INFCOM.2013.6566921*