Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.

ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-51

An Empirical Analysis of Finite Resource Impact on Cloudlet Performance in Mobile Cloud Computing

[Md Whaiduzzaman, Abdullah Gani and Anjum Naveed]

***Abstract*— In recent years, mobile devices such as smart phones, tablets empowered with tremendous technological advancements. However, due to limited resources, mobile device leveraging cloud computing by augmenting compute-intensive applications to the cloud. In spite of several advantages, still this distant cloud appears with several limitations such as latency and communication delay. Hence, cloudlet is a viable solution to augment the mobile device task to the nearest small scale cloud known as cloudlet. However, this cloudlet resource is finite which in some point appear as resource scarcity problem. In this paper, we highlight the cloudlet resource scarcity impact on overall performance in the cloudlet for mobile cloud computing. First we highlight the basic idea of cloudlet, applications, importance with present limitations. In addition, for empirical analysis, we make some definitions, assumptions and research boundaries. Moreover, we experimentally examine the finite resource impact on cloudlet overall performance. By, empirical analysis, with the several graphs, we explicitly establish the research gap and present cloudlet finite resource problem in mobile cloud computing.**

***Keywords*—mobile cloud computing, compute-intensive applications, resource scarcity, computation offloading, cloudlet**

I. Introduction

Cloudlet is a small cloud located in close vicinity to the mobile users connected through wireless communication. Cloudlet is installed on discoverable, localized, stateless servers running one or more virtual machines (VMs) on which mobile devices can augment resource intensive applications offloaded for computational resources [1]. It is a set of relatively resourceful computers that is well-connected to the Internet and is available for use by nearby mobile devices [2] .

Satyanarayanan, et al. [1] first introduced the cloudlet concept and mentioned it as a “data center in a box”. It is self-managing, requiring little power, Internet connectivity, and access control for setup. This simplicity of management make it feasible to use as a model of computing resources and to deploy on a business premises such as a coffee shop or a doctor’s office. Internally, a cloudlet resembles a cluster of multicore computers, with internal connectivity and a high-bandwidth wireless LAN for external access and having the virtualization capability of cloud computing. Hence, a cloudlet can be viewed as a surrogate or proxy of the real cloud, located as the middle tier of a three-tier hierarchy: mobile device, cloudlet, and cloud [3-5].

Mobile cloud computing liberates mobile devices from resource constraints by enabling resource-intensive applications to leverage cloud computing. Researchers named it as a cyber-foraging which can be realized using distant remote cloud. However, due to WAN latency, jitter, congestion, slow data transfer resulting increased power consumption and cost for user side [6, 7] . Hence, to alleviate these problems, clouds are being taken closer to the user by cloudlet concept. The benefits of utilizing cloudlet are the speed of service accessibility, the support of mobility, the enhanced application performance, the elongated battery life, and the reduced roaming data cost[8] .

Cloudlet has a major important role in Mobile cloud computing for its several distinguished advantages and features. Recently, researchers have found cloudlet as a viable solution for mobile cloud computing. We are going to mention them in this section [3, 9].Cloudlet and distant cloud have the same functionality with some differences. Cloudlet performs the tasks that are offloaded to the cloudlet using different offloading mechanism. Cloudlet has relatively higher resources compared to the mobile devices and effectively, task completion time is lesser on cloudlet, compared to the mobile device. However, unlike cloud where the user OS instance is stored along with modifications permanently, in case of cloudlet, the basic OS instances are available while user snapshots of the customized OS instances cannot be saved because of limited storage and lesser probability of reuse [10, 11].

Cloudlet have the same general architecture as a normal computer, having the virtualization capability , less powerful, smaller, and less expensive compare to distant cloud, which is ideal for role of a small scale servers installed in the public infrastructure. Cloudlet can be situated in common public area, business center, airport, coffee shops, shopping mall which facilitated the offloading facility to the mobile user by connecting the mobile devices as a thin client to the cloudlet [12, 13] .

There are several methods and offloading techniques introduced for application migration from mobile device to the computational clouds. One is VM migration, in which an already executing VM is suspended; its processor, disk, and memory state are transferred; and finally VM execution is resumed at the destination from the exact point of suspension. For application migration, Satyanarayanan, et al. [4] introduced a dynamic VM synthesis that enable mobile devices to deliver a small VM overlay to the cloudlet infrastructure that already possesses the base VM from which this overlay was derived. The infrastructure applies the overlay to the base to derive the launch VM, which starts the suspended execution of the suspension at the exact precise point [14, 15].

Md Whaiduzzaman, Abdullah Gani and Anjum Naveed
University of Malaya
Malaysia

II. Problem Analysis

Cloudlet has finite resource and this is an intrinsic problem of cloudlet which has a negative effect of its overall performance. In this study, we aim to highlight the resource scarcity problem and establish the impact of the resource scarcity on cloudlet performance by empirical analysis.

A. *System parameters*

We first define the resources, tasks and workload for cloudlet. Subsequently we define the performance parameters. Our defined several definitions for analysis are as follows:

Definition 1 (Resource): A resource represents an available unit which is required for executing a task. We can denote r as a resource and R can be denoted as a set of available resource.

Definition 2 (Cloudlet resources): For cloudlet, three types of resources are available such as CPU, memory and storage. The fundamental operation of most CPUs, regardless of the physical form they take, is to execute a sequence of stored instructions. In the context of performance, CPU investigations focus on power, throughput through the processor. This power is measured in million instructions per second (MIPS) [16]. Main memory is constantly occupied by several basic functions of the operating system, user data occupy the address space, and special segments are used for messaging or communication buffers and system buffers for certain system tasks [16]. Among Memory Parameters Memory\Available Kbytes is the amount of physical memory available to processes running on the server. Memory committed Bytes represent by the Memory\Committed Bytes and which is the amount of committed virtual memory. It should be less than 75% of uses [17].

Definition 3 (CPU Utilization or CPU uses time): CPU-usage or CPU time is the amount of time for which a central processing unit (CPU) was used for processing instructions of a computer program.

Definition 4 (Task): A task can be a logical unit of work which is executed by resource.

Definition 5 (User service time): User service time indicate the total time taken by the cloudlet and other related transfers time and network delay to deliver the computation service to the mobile user. User service time indicate the total time, including execution time and wait time, taken by the cloudlet to deliver the computation service to the mobile user.

Definition 6 (Throughput): Throughput is the number of tasks completed by the cloudlet per unit of time. It depends on several factors that can affect execution of a task [18]. Let a user application have 'n' tasks and they are submitted to run on 'm' machines from the Cloud provider. Let Te(n,m) be the execution time of n tasks on m machines. Let To be the time overhead due to various factors such as inter task communication delays and resource unavailability. Therefore, the total throughput of a cloud service is given by:

$\alpha = n/(Te\ (n,m)) + To)$

Definition 7 (Efficiency): The cloud system efficiency indicates the effective utilization of cloudlet resources and services. Therefore, a higher value of for efficiency indicates that the overhead will be smaller [1818].Hence, cloud system efficiency= Te (n,m)/(Te (n,m)+ To)

B. *Test bed*

In In this section, we describe experimental model, mobile client and cloud servers specification, communication infrastructure and data design. Data design including performance metrics, prototype applications, workloads, data generation process, and data collection apparatus are presented. Here, we analyze the impact and verify the severity of resource scarcity impact on cloudlet. Considering Open stack as a cloudlet service provider, we use mobile devices for a local user for the test bed.

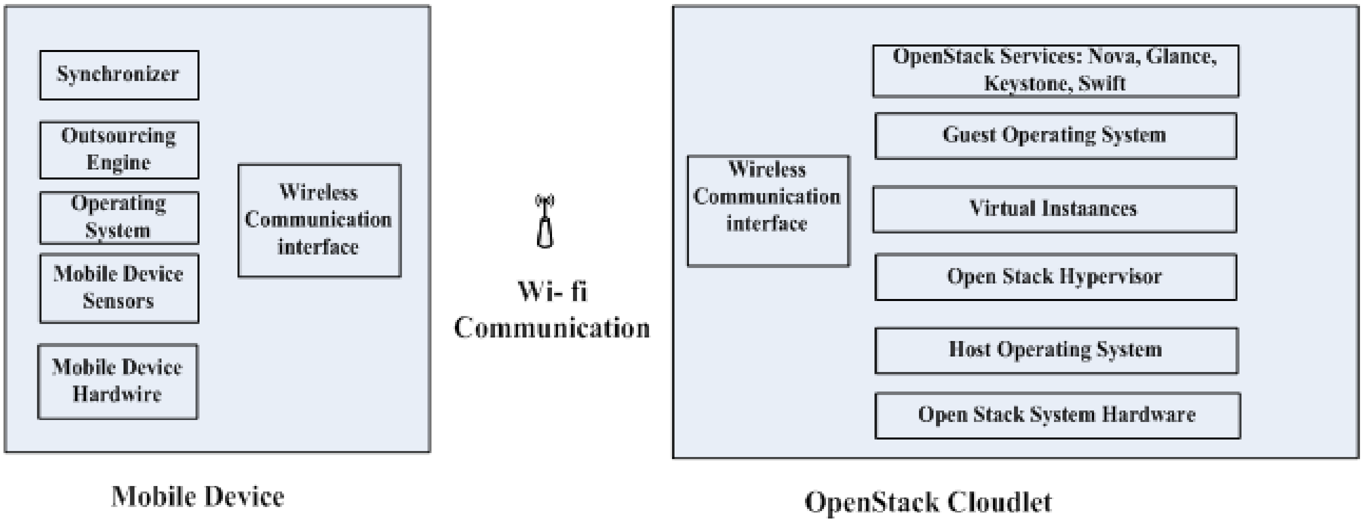

Figure 1. Schematic diagram of the Mobile device and Cloudlet

C. *Experimental Model s*

In this analysis, the two test bed modes are named as local mobile execution mode and remote execution mode specifically which is in cloudlet execution mode. We test our compute-intensive prototype application in these two modes.

TABLE 1. TECHNICAL SPECEFICATION OF MOBILE DEVICE USED IN EXPERIMENT

Mobile Device	Series	OS	OS Version	Processor	RAM	Chipset	Storage	Wi-Fi
Samsung Galaxy S2.0	GT-I9100	Android	V 2.3.4	Dual Core 1.2 GHz Cortex A-9	512 MB	Exynos-4210	16GB	802.11 a/b/g/n

In local execution mode, all the application components are executed locally inside mobile device, whereas in the intensive component of the prototype mobile application is executed as a cloud-based resource. Our graphical representation of the Experimental model is depicted in Figure. 1 and its components are described as follow.

Mobile Clients : The mobile devices used in this experiment are Samsung Galaxy S2 GT-19100G, featuring Dual-core 1.2 GHz Cortex-A9 processor, 512 MB RAM, Wi-Fi 802.11 a/b/g/n, running Android v2.3.4, Chipset Exynos 4210, GPU Mali-400 , Sensors-accelerometer, gyro, proximity, compass , and with GPS, MIPS 750.

TABLE 2. TECHNICAL SPECEFICATION CLOUDLET

Cloudlet Platform	Version	Original OS	Processor	MIPS	VCP	RAM	Storage	OS
Open Stack	Havana	Linux UBUNTU 12.04	2.4 GHz Xeon	32000MIPS	4 core	8GB	80GB	Cirros3.03

Cloud-based Resources: In this analysis, we built our server side using the following components. On top of the bared metal hardware and host operating system in cloud, several layers and components exist which are explained below.

Wired Communication Interface (WCI): Wired Communication Interface (WCI) is the communication interface of the cloud VM. The client request is delivered to the server VM via this interface point.

Hypervisor: Hypervisor or VM manager is a cloud-side application that manages the creation, execution, and destroying of the VMs. Maintaining cloud services is feasible via Hypervisor only. Hypervisor builds a layer over the host operating system to provide virtual processing infrastructures to the mobile service consumers.

Virtual Machine Instance: In order to provide computing powers in cloudlet, we utilize VM instances featuring varied computing powers and configurations follows. The cloudlet server located in close to the mobile device featuring 4 VCPU cores, 8 GB RAM, having 2.4 GHz Xeon processor with MIPS 32000, storage 80 GB running on Linux Ubuntu Server. GCC compiler in C program version 4.8.3. The workload of the program instruction size is 120X 109 and executable code size is 8,47 KB. For embedding the library file to run and execute the application, we compiled it as a static by differentiating the architecture r as an arm processor. In the cloudlet size, we used Virtual machine as described in Table with the resource2

Data Collection Tools: Data collection tools are designed carefully to minimize the man-made mistakes. Data of time nature are collected using our own designed software automatically. Program start before execution and ends after ending the task.

III. Cloudlet finite resources capacity and its overall impact

In this empirical study, we study and analyze the cloudlet resources and investigate the overall impact of cloudlet performance. First, start with the core resource of cloudlet which is CPU. Mainly we consider the compute-intensive applications for our investigation and we designed our workload accordingly. CPU is the primary resource of any compute unit and CPU power is a critical factor for those applications, which are strongly CPU bound, such as scientific and technical applications with long sequences of mathematical calculations. We mainly focus on CPU utilization and its effect with the increase of workload constantly with a certain time interval.

A. *CPU Utilization*

As we defined the CPU Utilization or CPU uses time: as is the amount of time for which a central processing unit (CPU) was used for processing instructions of a computer program. Normally, CPU utilization should not exceed 75% for any application . For our experiment, we increase the workload and observe the CPU utilization by the following graphs and Figure2: Workload and CPU utilization scenario from the

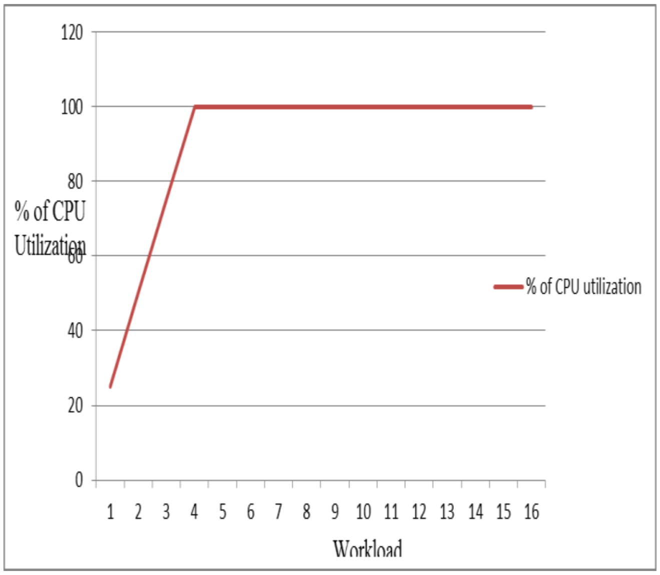

Figure 2. Workload and CPU utilization

Fig2, we examine the CPU utilization behavior or pattern of the CPU. We can confirm that up to 4 workload, CPU utilization reaches it's peak 100% and from the workload 5 onwards it remains 100% CPU utilization. We investigate the effect of continuous increasing workload with the two variations:

(a) If we increase the resources, in this case CPU core, like 8 0r 12 then up to 8 or 12 workloads it can serve below 100% utilization. After that, in case of 8 core, it remains 100% after 8 workload and for 12 core, it remains 100% utilization constant with the increasing load behavior

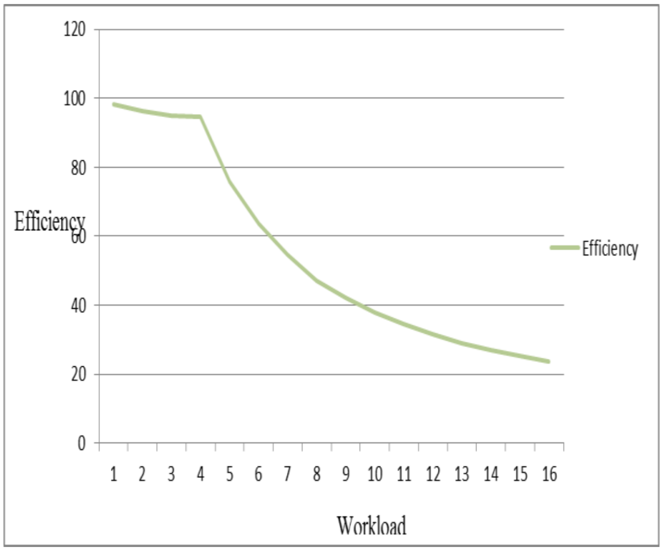

Figure 3. Workload and Efficiency graph

(b) Another case, if the CPU capacity or strength replaced by the stronger CPU, then it reaches it's 100% with in shorter time compare with the comparatively weaker CPU, if other conditions remain same for the experiment.

In this experiment, we can identify the CPU utilization pattern and CPU resource scarcity of the cloudlet. We cannot see the impact of increasing workload after the utilization reaches 100%.. Let us consider the overall efficiency to find out the effect of the increasing workload of the cloudlet remaining the current situations and conditions unchanged

B. *Efficiency*

For the cloudlet, efficiency is defined by the time taken to execute a task divided by the time including the overhead time with the same task. Therefore, we again bring the equation as follows,

Efficiency= Te/(Te + To+ Ts)

So, Efficiency=Time taken for single application divided by total time.

Therefore, cloud system efficiency= Te (n,m)/(Te (n,m)+ To)………equation (1)

Our experimental results are presented in Fig. 3. From the Figure. 3, we can say that efficiency is decreasing with the first few loads and after that it went down exponentially. This means that overall execution time increases but the task finishing time for each individual task is remaining same. To explain more to the effect to the cloudlet, now we need to investigate the overall user service time effects with the workload increases. This also gives us idea of how cloudlet performance affects the users

C. *User Service time*

In cloudlet, user service time defines by the total time needed to serve a user including transfer time to the cloud , execution time and the cloud and other relevant delays that altogether affects the user is known as user service time.

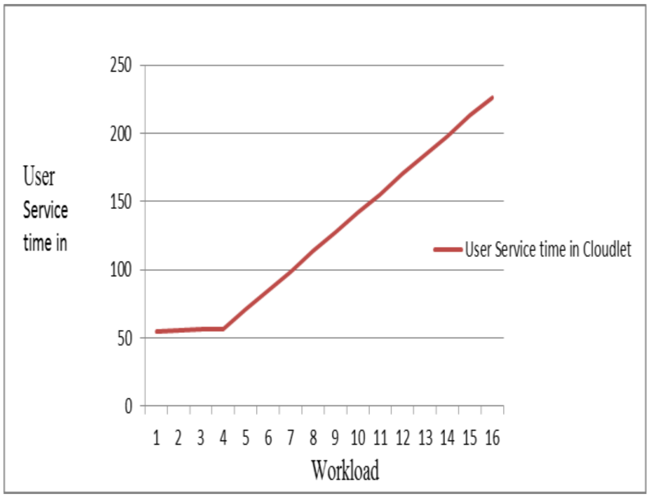

Figure 4. Workload and User service time graph

Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-51

Service time increase with the increasing of workload. Therefore, it overall plays a negative effect on user service time which make user to wait a longer time to get the result of their offloaded computation from the cloudlet which indicate the poor user experience of cloudlet. If we increase, the number of CPU core, it remains steady with the number of core and after that increase sharply with the increasing number of load increase constantly with the time. Another case, stronger CPU takes lesser time to execute the workload but it remains overall the same effect.

In short, we can examine and summarize from the aforementioned empirical experiments that finite resources of cloudlet make it resource scarce and that impact negatively of cloudlet overall system performance. The cloudlet performance affects the user response time and in the end user might not find submitting the tasks to cloudlet as useful. It may be noted that if we can somehow reduce the workload, the service time of users can be improved.

IV. Conclusion

In this paper, we highlight the cloudlet resource scarcity impact on overall performance in the cloudlet for mobile cloud computing. First we discuss the basic idea of cloudlet, types, applicability, importance and suitability with present limitations. Then, for empirical analysis, we make some definitions, assumptions and research boundaries. In addition, we experimentally examine the effects and impacts of finite resource on cloudlet overall performance. Among the several graphs, we explicitly establish the research gap and present cloudlet finite resource problem. In future, we will come out with a solution which having a framework for resource limited cloudlet problems by enhancing the cloudlet resources in mobile cloud computing.

Acknowledgment

This work is fully funded by the Malaysian Ministry of Education under the High Impact Research Grant of University of Malaya UM.C/625/1/HIR/MOE/FCSIT/03.

References

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," Pervasive Computing, IEEE, vol. 8, pp. 14-23, 2009.

[2] A. Bahtovski and M. Gusev, "Cloudlet Challenges," Procedia Engineering, vol. 69, pp. 704-711, 2014.

[3] M. R. Rahimi, "Exploiting an elastic 2-tiered cloud architecture for rich mobile applications," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, 2012, pp. 1-2.

[4] M. Satyanarayanan, G. A. Lewis, E. J. Morris, S. Simanta, J. Boleng, and K. Ha, "The Role of Cloudlets in Hostile Environments," IEEE Pervasive Computing, vol. 12, pp. 40-49, 2013.

[5] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," Journal of Network and Computer Applications, vol. 40, pp. 325-344, 4// 2014.

[6] S. Bohez, T. Verbelen, P. Simoens, and B. Dhoedt, "Allocation Algorithms for Autonomous Management of Collaborative Cloudlets."

[7] E. Ahmed, A. Akhunzada, M. Whaiduzzaman, A. Gani, S. H. Ab Hamid, and R. Buyya, "Network-centric performance analysis of runtime application migration in mobile cloud computing," Simulation Modelling Practice and Theory.

[8] M. Whaiduzzaman, M. N. Haque, M. Rejaul Karim Chowdhury, and A. Gani, "A Study on Strategic Provisioning of Cloud Computing Services," The Scientific World Journal, vol. 2014, p. 16, 2014.

[9] A. Gani, G. M. Nayeem, M. Shiraz, M. Sookhak, M. Whaiduzzaman, and S. Khan, "A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing," Journal of Network and Computer Applications, vol. 43, pp. 84-102, 8// 2014.

[10] H. Qi, M. Shiraz, A. Gani, M. Whaiduzzaman, and S. Khan, "Sierpinski triangle based data center architecture in cloud computing," The Journal of Supercomputing, pp. 1-21, 2014/04/26 2014.

[11] M. K. Nasir and M. Whaiduzzaman, "Use of Cell Phone Density for Intelligent Transportation System (ITS) in Bangladesh," Jahangirnagar University Journal of Information Technology, vol. 1, 2012.

[12] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," Future Generation Computer Systems, vol. 29, pp. 84-106, 2013.

[13] M. Shiraz, M. Whaiduzzaman, and A. Gani, "A study on anatomy of smartphone," J Comput Commun Collab, vol. 1, pp. 24-31, 2013.

[14] M. Whaiduzzaman, A. Gani, N. B. Anuar, M. Shiraz, M. N. Haque, and I. T. Haque, "Cloud Service Selection using Multi-Criteria Decision Analysis," The Scientific World Journal, 2013.

[15] M. Whaiduzzaman and A. Gani, "Measuring security for cloud service provider: A Third Party approach," in Electrical Information and Communication Technology (EICT), 2013 International Conference on, 2014, pp. 1-6.

[16] W. W. Osterhage, Computer Performance Optimization: Springer, 2013.

[17] N. J. Gunther, Analyzing Computer System Performance with Perl::PDQ, Second Edition ed. Springer Heidelberg Dordrecht London New York: Springer.

[18] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," Future Generation Computer Systems, vol. 29, pp. 1012-1023, 6// 2013.