TOWARDS ENHANCING RESOURCE SCARCE CLOUDLET PERFORMANCE IN MOBILE CLOUD COMPUTING

Md Whaiduzzaman[1], Abdullah Gani[2] and Anjum Naveed[3]

[1,2,3]Center for Mobile Cloud Computing Research (C4MCCR)
Faculty of Computer science & Information Technology,
University of Malaya, Kuala Lumpur, Malaysia
[1]wzaman@ieee.org,[2]abdullah@um.edu.my,[3]anjumnaveed@ieee.org

ABSTRACT

In recent years, mobile devices such as smart phones, tablets empowered with tremendous technological advancements. Augmenting the computing capability to the distant cloud help us to envision a new computing era named as mobile cloud computing (MCC). However, distant cloud has several limitations such as communication delay and bandwidth which brings the idea of proximate cloud of cloudlet. Cloudlet has distinct advantages and is free from several limitations of distant cloud. However, limited resources of cloudlet negatively impact the cloudlet performance with the increasing number of substantial users. 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 analyse the cloudlet resource scarcity problem on overall performance in the cloudlet for mobile cloud computing. 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, we explicitly establish the research gap and present cloudlet finite resource problem in mobile cloud computing. In this paper, we propose a Performance Enhancement Framework of Cloudlet (PEFC) which enhances the finite resource cloudlet performance. Our aim is to increase the cloudlet performance with this limited cloudlet resource and make the better user experience for the cloudlet user in mobile cloud computing.

KEYWORDS

Mobile Cloud computing, Cloudlet, Resource Scarcity, Performance Enhancement.

1. 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

David C. Wyld et al. (Eds) : CCSIT, SIPP, AISC, NLP - 2015
pp. 01–11, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.50401

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. [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 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].

To realizing the Cyber foraging using a cloudlet, the VM-based cloudlet concept has recently evolved to component-based cloudlets consisting of a group of computing nodes (both fixed and mobile) that are sharing resources with one another. Software components on the mobile device can then be redeployed at runtime to other nodes in the cloudlet according to some optimization criteria, such as the execution time, energy consumption and throughput[7, 10] . These applications are involving multiple users interacting with each other in a real-time fashion. The resource-sharing concept of component-based cloudlets opens a promising research area for collaborative applications which not only sharing computing resources but also share user data such as processing results and context information [9, 15]. Cloudlet has a major important role in mobile cloud computing for its several distinguished features as follows [11].

The mobile users get instant direct access to the cloudlet, due to the close proximity of the user and the cloudlet which eliminates several drawbacks introduced by the communication latency, jitter, and slow data transfer of cellular network. The conventional benefit of offloading the computational resource intensive tasks into cloud is still preserved in cloudlet in which the mobile device can get rid of resource starvation. Finally, since cloudlet is the near vicinity of the mobile user, it can save money by avoiding expensive data charging in roaming situation [8, 16]. Hence, 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[13].Section 2 explains the problem analysis of resource scarce cloudlet, Section 3 describes the PEFC framework with different components, and Section 4 presents the significance of the framework and finally Section 5 draws conclusive remarks with our future research direction.

2. 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.

2.1 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.

Definition 3 (Task): A task can be a logical unit of work which is executed by resource.

Definition 4 (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 5 (Remote application execution time): The total time is taken for execute the application in cloudlet

Definition 6 (Local application execution Time): The time is taken to execute the program in local mobile device.

Definition 7 (Large application/program): We purposely make this application as a compute-intensive which comparatively take longer time than our small program.

Definition 8 (Small Application/program): We purposely make this application as a compute-intensive which comparatively take lesser time than our large program.

2.2 Experiment

In this section, we describe experimental model, mobile client and cloud servers specification, communication infrastructure and data design. 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.

One is constrained of resources of cloudlet, since these resources are free to use, therefore the available resources are not adequate when the number of users or applications request for computational services from the cloudlet. The increasing user load, in some point make the cloud resource scarceness and ultimately the resource constraint of cloudlet take more time to application execution, which affect the user experience with long application execution delay. It also creates the problem of on demand resource availability by the new user request since resources are provisioned already and lack of available resources for the new requests. These effects severely hindered the main purpose of using cloudlet.

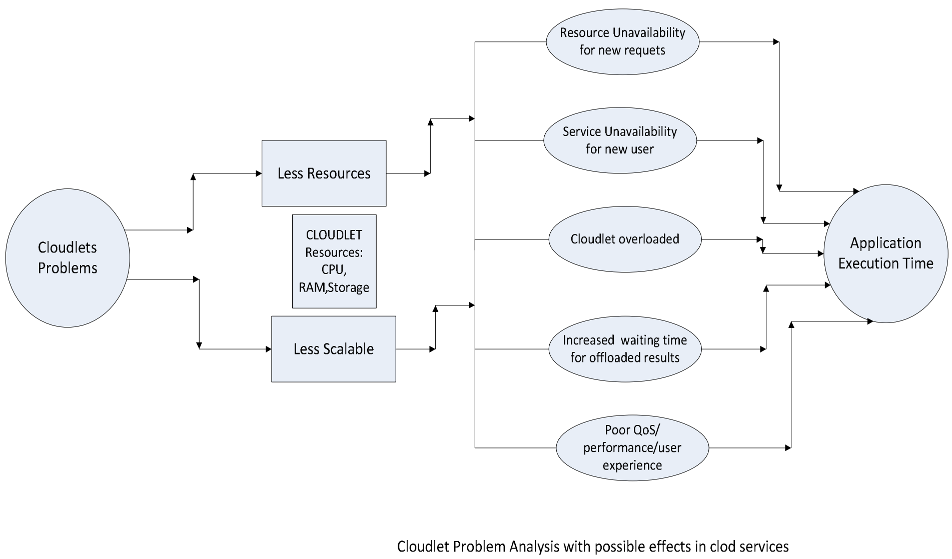

Figure 1. Cloudlet problem Analysis with possible effects in application execution time

If we consider the aforementioned problems in cloudlet which are fewer resources and less scalable, ultimately it affects the application execution time. Among the resources in cloudlets, for our experiment we consider the processor speed, and how effective it will be to execute the application in cloudlet. In application execution, we consider two aspects: One is in remote application execution time which should be in cloudlet, and other one is in local application execution time which we consider the time taken by the local execution by the mobile device. For our experiment, we define and design a large program and a small program to differentiate it by two types of execution time. It is obvious that large program computation time is more than the small program computation time. We made it purposely to done our experiment. Several important terms for this experiment:

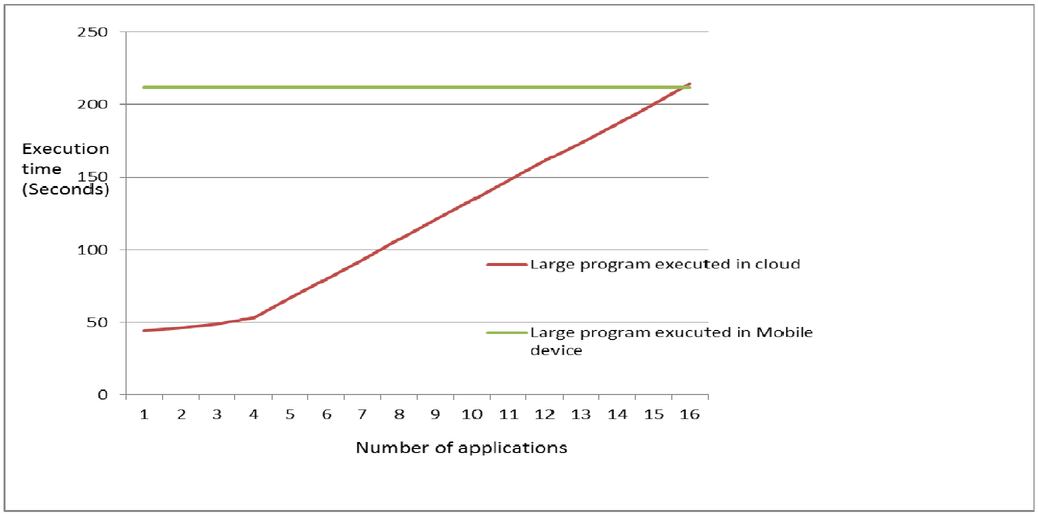

Figure 2. Comparison of small program execution time in cloudlet and mobile device

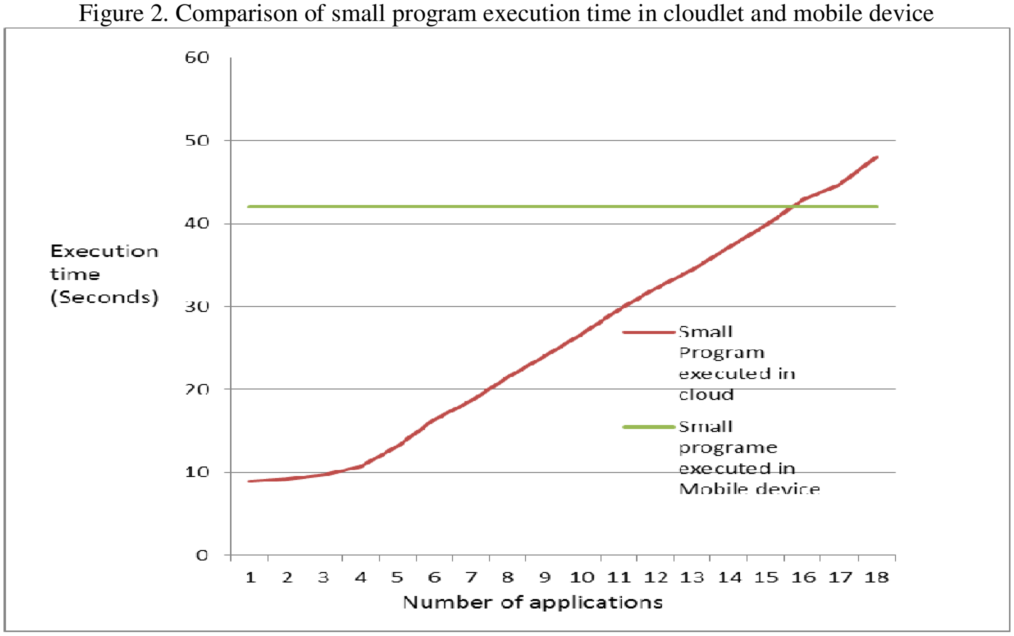

Figure 3. Comparison of Large Program execution time in cloudlet and mobile device

We have done this experiment in Open stack Havana Cloud running in Ubuntu12.04 Linux server and Samsung Galaxy S2 is used as a mobile device in the lab environment real test bed.

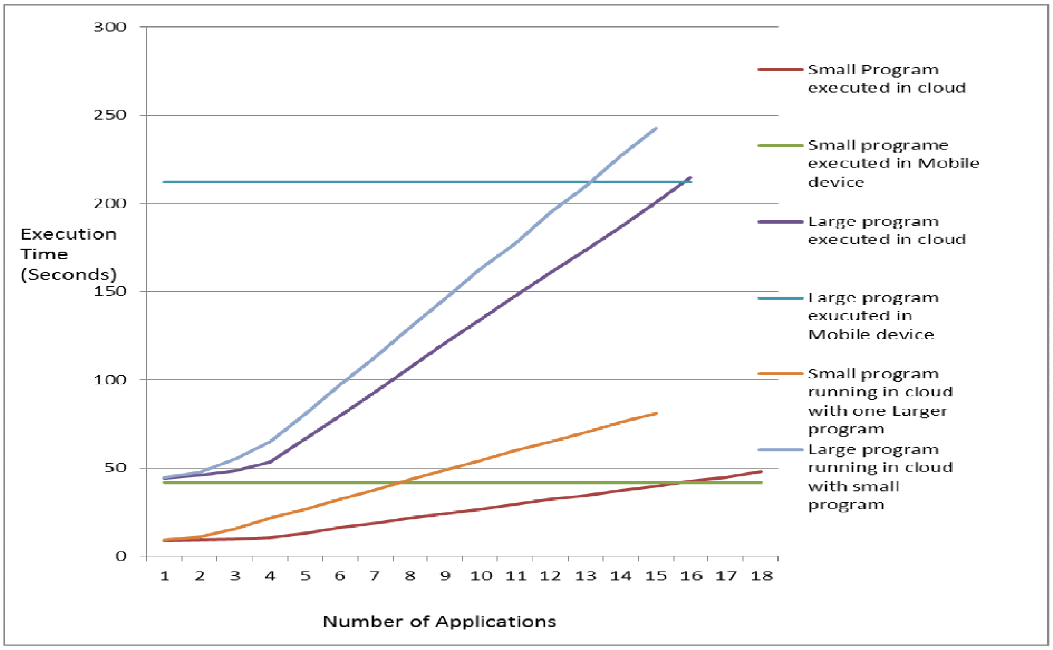

Figure 4. Comparison of different types program execution time in cloudlet and mobile device

From Experiments and Figure 2,3,and 4 , we can confirm that all of the cases if 16 users using the cloudlet, then 17th number user have no benefit from the cloudlet in terms of program execution time in cloudlet. Because, in this case, the mobile device can run faster, if execute in locally.

3. PERFORMANCE ENHANCEMENT FRAMEWORK FOR CLOUDLET

This section reports a Performance Enhancement Framework for Cloudlet (PEFC) for MCC. The objective is to explain and address the issues of resource scarcity problems which hindered the cloudlet performance. This section explains the architecture and operating procedure and performance of the proposed framework.

This framework basically consists with two major building blocks. One is the user side; we assume it a mobile device. In this study, we assume all the devices are same type in operation perspective. In the cloud side, we call it cloudlet, a small scale cloud which is built and operated in open source cloud software named as open stack. We propose a novel Performance Enhancement Framework for Cloudlet (PEFC) for MCC. PEFC address the issues of resource scariness in cloudlet by offloading and shifting some of its workload or process to the nearby mobile device using the Wi-Fi communication and finally sent the processed output to the specific user. To design and development of the framework, we consider several important offloading aspects. For decision making, we employed several decision making algorithms and we consider the mobility pattern for the framework. Two tiers architecture of our framework basically builds up with the mobile user which we consider as an end user. This user can be identified as a smartphone, tables and even the laptop user.

After user sending the tasks to the cloudlet, then the second part is executed by the cloudlet for task completion. This step involves several serious consideration and efficient consecutive process for the whole task completion cycle. First of all, the framework check the usability of the cloudlet whether it is practically beneficial for the user to offload the task. If yes then it goes through several sections instructed and directed by our algorithms. We explain the each steps individually the main job and their internal execution to complete the whole task and send back the result to the user. We present the framework components and building blocks in Fig 5. The following section explains the components of PEFC framework:

3.1 Task Handler

Task handler is situated in Cloudlet side which first receives the task request from the mobile device. Task handler mainly analyzes the task and makes a weight for the computational benefit from the cloud. This task handler initially assesses the whole task and the priority and importance. For the further analysis then it send to the aggregator and profile section.

3.2 Aggregator

The aggregator mainly aggregates all the components of the task and reorganizes and rearranges it according to the sequence of the application. It basically incorporates the different components of the task and reshuffles and reorganizes the random part to make it in an orderly meaningful manner.

3.3 Profiler

Profiler mainly responsible for analyze the applications and its different components. It describes the need of demand for computation units need to complete the task execution. In addition it can make the all components of profile to identify and reorder the task and calculate the whole necessary resources needed to finish the work.

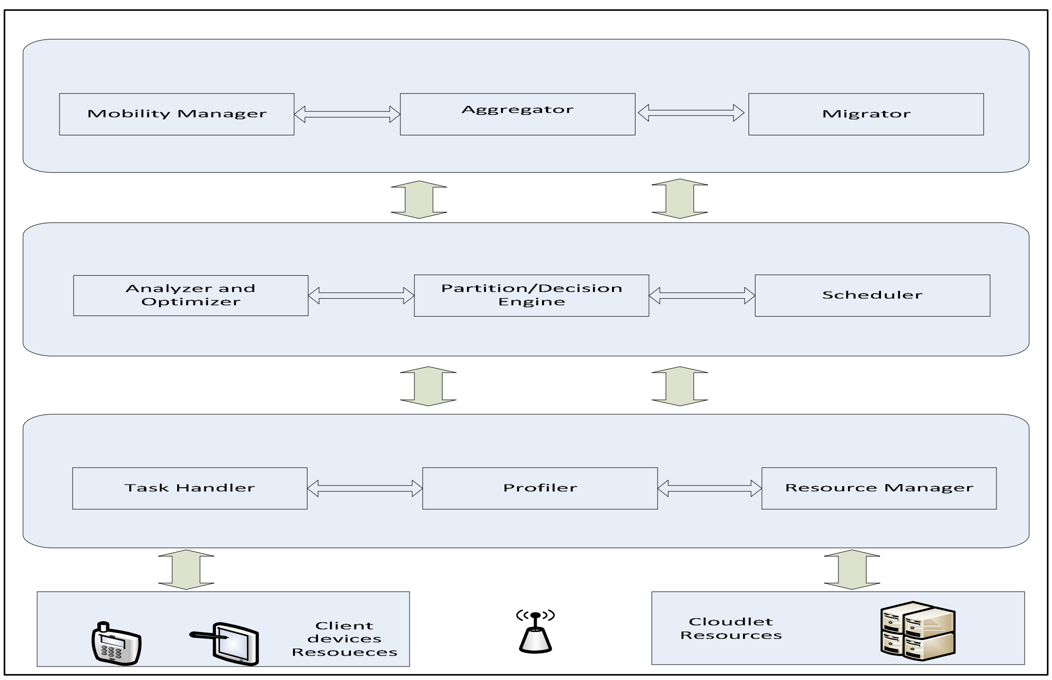

Figure 5. Main building blocks of Performance Enhancement Framework for Cloudlet

3.4 Analyzer and Optimizer

After profiling, tasks are handed to the analyzer and optimizer section for further analysis based on the profiler data. By using optimization techniques, analyzer and optimizer decide to further processing of the request tasks which should be optimally done by the available resources for the cloudlet and mobile devices.

3.5 Scheduler

Scheduler makes the scheduling for complete the task. It assigns the priority which devices the task will reschedule and overall available resources for task scheduling. When task finish then it update accordingly and make the resource free for the other sequential important task for completion.

3.6 Partition/Decision Engine

The decision engine makes the decision of application execution. It could be the in to cloudlet or to the mobile devices other than the sending devices. This section makes decision after analysing the resources associated with the mobile devices resources. We employ several decision making algorithm in this stage to get the optimized and enhanced performance by make the decision that should be the optimized one.

3.7 Mobility Manager

Mobility manager hold the status of all the mobile devices connected to the cloudlet in the proximity of cloudlet. It can be done by storing all the Wi-Fi signal strength registered with the Wi-Fi zone and their signal strength confirm us the proximity of the all the nearest devices. Among them, from the client profiler, we can have the resources latest update and the information who are waiting for the cloudlet services by the receiving the results from the cloudlet. Mobility is an important aspect which we also consider in our framework. From the mobility pattern analysis, when decision engine make decision which mobile device, it is going to offloaded task to the client devices can help the predicted mobile device.

3.8 Resource Manager

Resources indexing are keeping as a database to the cloudlet. All the available and presently used resources are keep track by the resource manager. Therefore resource manager is responsible for keep all the updated data and information in the cloudlet which is very important for the decision engine to estimate the available resources from the cloudlet. Hence, every new device joining in the network should register their resources by the profiler to the cloudlet and again, when any device leave the network, instantly the resource manager remove the device and it's available resources from the database.

3.9 Migrator

Cloudlet use Migrator to transfer the portion of data or code segment or process to migrate to the surrounded mobile devices. Obviously, decision engine using our algorithms make decision to choose the best case mobile device. After that for sending the code, migrator is responsible of sending to the mobile device. Migrator receives the result from the mobile device to process it again by sending to the aggregator and other further steps.

3.10 Client Devices Resources

The resources which are embedded with the client mobile devices are mentioned or marked here as user device resources. It could be CPU, memory or even storage. Several cases it could be the installed software that cannot be processed by the cloudlet or clod let has not installed with the software facility. We can have huge sensors now a day that could be great resources as well such as GPS, camera, thermometer, location apps, and embedded other latest sensors.

3.11 Cloudlet Resources

Resources which are belongs to cloudlet are normally considered as CPU, memory, and hard disks. In this experiment and our frame work, mainly we consider the compute-intensive applications, hence we identified the CPU is the main resources. As, we proof in chapter 3 that certain cases, this resources are not adequate and we call it some point it as a resource scarcity which degrade the overall cloudlet performance negatively. We aim to consider the problem and propose the solution to shift some application or process from the cloudlet to the nearest mobile devices that they can act as collaborative resources for the cloudlet. Since when the loads are reduced from the cloudlet, ultimately the performance has been increased and the results we get from the nearest mobile devices as working as a resource node or hub eventually make the whole process faster and make the cloudlet to increase the performance and can get rid of resource scarcity.

4. ADVANTAGE AND SIGNIFICANT

Our proposed framework has several important and significant features and charters tics which we outline below:

4.1 Enhanced Resources

One of the most significant features of this framework is it eventually increase the resources of cloudlet. In fact, practically, it does not increase the physical resources of the cloudlet .However; it does ultimately the same thing, if we increase the physical resources to the cloudlet. Especially when it dealt with the shifting tasks from cloudlet to the mobile device, in reality, it reduces the load from the cloudlet resources which free the cloudlet resources.

4.2 Nearest Proximity

The resource rich mobile devices are located to the close proximity of the cloudlet which brings the mobile devices additional or extra resources nearby to the cloudlet. Moreover, this nearness makes the framework more stable because it is less affected by the band width or network latency which is common for distant cloud.

4.3 No Cost

In the cost perspective, no costs are involved with this model and we know the cloudlet service is free. Hence this model is pretty straight forward no payment module involve with in it. This also help easier to implement the model and encourage the near vicinity of the cloudlet user to use their resources as free but still the mobile user who give his/her resources for the cloudlet ensure the advantage of time or energy benefit. This establishes the strong feasibility of free service and in this model the cloudlet and the mobile user both are getting mutual benefit with this model.

4.4 Performance Enhancement

Most important aspect of this framework, we get extra resources without any payment which ultimately increase the overall performance. It works both ways, at the same time it ensure the cloudlet performance increase and well as the mobile user performance by helping the mobile user to perform his tasks in timely manner with time and energy benefit.

4.5 Network delay

Our proposed framework considers the resources from the cloudlet and some are from the nearest mobile devices collaboratively. Therefore, this framework model do not need cellular or mobile network for communication between client and cloud. Moreover, it works using the wireless network with Wi-Fi connection, hence notably important that this proposed framework is free from communication latency and network delay.

4.6 Adaptive ness

To make the experiment simple and to avoid complexity, we make the framework simple and use one type of mobile devices. However, this model is suitable and appropriate for other devices with different operating systems. In short, it is feasible and supports the heterogeneity devices and operating system. Therefore, this framework has a strong adaptive ness with different platform and software which make it as strong business feasibility in the marketplace for its adaptive ness.

5. CONCLUSIONS

In this paper, we highlight the cloudlet resource scarcity impact on overall performance in the cloudlet for mobile cloud computing. First we explain several basic concepts of cloudlet, then, for empirical analysis, we make several assumptions and research boundaries. In addition, we experimentally examine the effects and impacts of finite resource on cloudlet overall performance. We establish the research gap and present cloudlet finite resource problem. In this study, we propose a framework, PEFC, to enhance the cloudlet performance. Our proposed framework, mitigate the resources scariness of cloudlet by shifting the load to the nearest idle mobile device to enhance the cloudlet performance. We describe the frameworks several components and highlights it's the important and distinct features. Finally, our framework shows the cloudlet performance enhancement and in our future work, we will implement the framework and empirically investigate the performance enhancement of the cloudlet in mobile cloud computing.

ACKNOWLEDGMENTS

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] M.Whaiduzzaman, A.Gani, N.B.Anuar, M.Shiraz, M.N.Haque, and I.T.Haque, "Cloud Service Selection Using Multicriteria Decision Analysis," The Scientific World Journal, vol. 2014, p. 10, 2014.