International Journal of Intelligent Computing and Technology
Vol. (1), No. (1), 2017, pp. 41-47

Novel Framework for Elastic Resource Allocation in Mobile Cloud Computing

P. Kannan[1], M. Durairaj[1]

Research Scholar, Assistant Professor,

School of Computer Science, Engineering and Applications,

Bharathidasan University, Tiruchirappalli, India

kannancsbdu@gmail.com, durairaj1.bdu@gmail.com

ABSTRACT

Mobile cloud computing is an important emerging area where mobile users utilize services such as Infrastructure, Software and Platform in an on-demand elastic manner. It provides resources based on mobility, bandwidth, network coverage, cloudlet location and resource offloading. Elastic resource allocation dynamically disintegrates; distribute resources to mobile cloud users. It is one of the major challenging tasks in mobile cloud environment. Hence, this paper proposes a novel framework for elastic resource allocation for mobile cloud users. This work highlights the techniques involved in the elastic resource allocation with likelihood is important to handle cloud computing resources in an educational cloud for university.

Keywords: *Mobile Cloud computing, Elasticity, Resource management, Partitioning.*

1. INTRODUCTION

Cloud computing is a new business model focusing on resource-on-demand, pay-as-you-go and utility computing [1]. Mobile Cloud Computing (MCC) is the computing, which refers to anytime, anywhere access to applications and data through the internet from standalone devices and mobile devices. In MCC, resources are accessed on-demand basis which resulted in reduction of cost investments for users by hiring resources rather than purchase them. The user can access the applications and data from anywhere at any point of time through internet.

Existing elastic application models for cloud are not well suited for mobile cloud due to scalability, mobility, network latency, cost and maintenance. As a result, new experiments and proposal of new framework models for mobile cloud are emerging. The ultimate aim is to render better use of device resources with scalable, minimum latency and low cost. Elasticity [2] is an essential feature in cloud computing. Instead of fixing the system resource allocation in advance, the cloud system dynamically responds to actual variations by applying virtual resources.

Partitioning [3, 6, 9] of resources is a key factor in elasticity which dynamically adds or removes virtual resources according to the variations of workload. To accomplish consistent migration and suitable offloading [4, 5, 8] between mobile devices and server, each application should be divided into an elastic module called

Weblets [7]. While partitioning, we should consider resource consumption and data dependency to avoid data breakage.

The partitioning technique can be done in two ways.

- Offline or static partitioning
- Online or dynamic partitioning

Offline or static partitioning: This technique applies on a fixed count of partitions. Here, applications are required for two stage processing. First, Mobile user needs a user interface on mobile devices. Second, computational modules of the application are transferred to suitable cloud server.

Online or dynamic partitioning: In this technique, partitioning and integration involves in a dynamic way of processing loads of mobile devices and mobile cloud server [13].

Application of partitioning technique plays a significant role in elastic application implementation in mobile cloud computing. Partitioning the application disintegrates the modules and process concurrently. Ioana Giurgiu et al. suggested two step approaches for optimal partitioning the application between mobile devices and cloud server. Initially, they abstracted the workflow of an application as a data flow graph of many interconnected software modules. To find the optimal cut off by applying a partitioning algorithm, the output of the first level is considered for the second level. For application partitioning, they have proposed ALL and K-step algorithms [10]. At first, ALL algorithm does partition for offline conditions based on network conditions. Later, K-step algorithm works in an online environment.

AlfredO framework was proposed by Giurgiu et al. to disintegrate and distribute the logic and application layer of elastic application [11]. Here, the data layer presents are in cloud server and the user interface is on client side. The key factors like data transfer, response time and cost between mobile devices and server are considered to formulate an elastic application middleware to different layers of an application by partitioning automatically.

The application partitioning technique is statically partitioned in MISCO [12] framework based on map reducing. At first, this map function gets a key/value pair and produce key/value pairs to process to the next level. Next, this function joins all the key/value pairs with the existing key to produce final output. Furthermore, this framework contains three major roles master, mappers and reducers. Master works as the organizer for resource management, scheduling tasks, etc. Mappers read input data which are divided into blocks through DFS/IO and produce intermediate key/value pair which will be stored in its LFS (local file system). The address of intermediate output will be sent to masters, who order to reducers to get the intermediate results as reducer's input. The reducer then reads the data from mappers by using RPC (Remote Procedure Call).

Messer [14] et al. launched a framework which is designed for application load balancing of cloud server and mobile devices such as PDAs, laptops and smartphones. This framework contains partitioning component, offloading technique, etc. Partitioning component disintegrates the application by complying partitioning policy. In offloading technique, application profiling factor gives the feasible solution. It monitors the execution history of offloading application as well as predicting the application resource requirement for the future. Byung-Gon and Petros [15] proposed applications by using dynamic partitioning technique between

clouds and mobile devices. They mentioned that the static partitioning technique in applications does not provide any best optimal solutions at the user level because most of the applications are used at different level by different users to access the cloud. They recommend that the execution environment and workload could be considered while taking decisions in partitioning.

The remaining of the paper is formed as follows: The proposed model for partitioning the cloud application for doing collaborative and interdisciplinary research activities of academic institutions is discussed in Section 2. Results and discussions is shown in Section 3. Finally, Section 4 concludes this paper based on the proposed work.

2. METHODOLOGY

As partitioning technique [17] divides an elastic application into modules, here the proposed framework helps to enhance elasticity and research activities at the university level. This framework is mainly classified into four layers such as Elastic Handler, Execution Handler, Resource handler, Partitioning layer and Cloud users. The different layers and entities involved in the elastic resource allocation framework are illustrated in Fig. 1.

Elastic Handler: This handler carries out elasticity based on prediction and then it transfers the control to the execution environment like mobile devices or to any device which access cloud data. It acts as an intermediate between partitioning layer and Execution environment layer. Data is synchronized by sending during this process. Moreover, Elastic handler retransmits the data if there is any loss at the time of transmission.

Execution Handler: Traditional access methods use client-server computing. In Cloud computing, execution handler distributes the resources on the server. It monitors and decides the best execution environment which is suitable for elasticity. In addition to that, this layer traces behavior of the applications for recovering the resources.

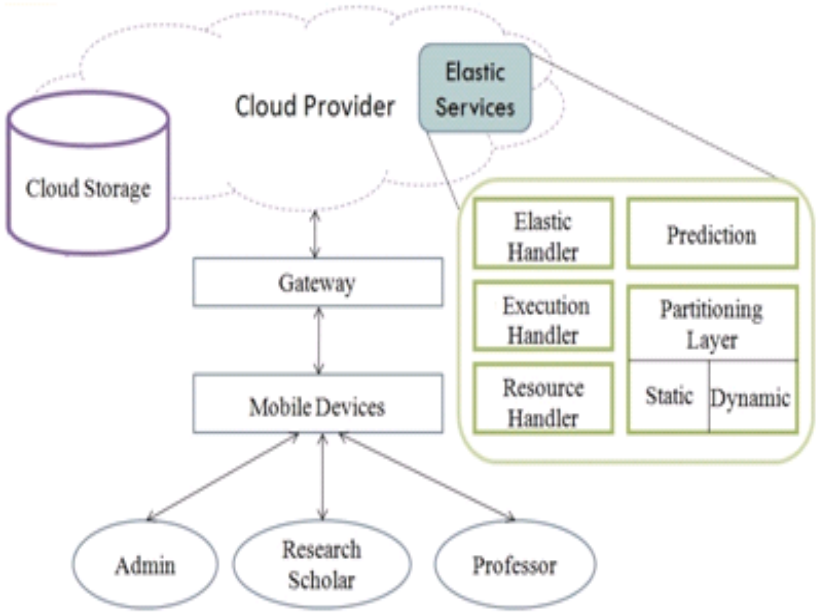

Fig. 1 Framework for elastic resource allocation

Resource handler: It decides the percentage of resource which is to be executed in mobile devices or cloud servers after partitioning based on the resource availability such as network connectivity, location, battery

consumption and node availability, etc. Let's assume the application has 80% possibility of partitioning, then this layer divides the application and it offloads the partitioned application to the cloud server.

Partitioning Layer: This layer plays a significant role to predict the behavior of resources based on the concept of static and dynamic of its nature. This layer partitions the application based on the requirement of elasticity. The applications are divided into as Elastic Partitioning Container (EPC) which traverses between mobile devices and cloud servers.

$$EA = \sum_{i=1}^{n} epc_n \quad (1)$$

Here, EA indicates resource utilization function. EPC is elastic partition container for module i. The application in the cloud is partitioned up to level n to produce results to client as expected.

Cloud users: This layer focus on how the user interacts with cloud to access data by using thin client with internet connectivity. The cloud user may be a research scholar, admin and professor.

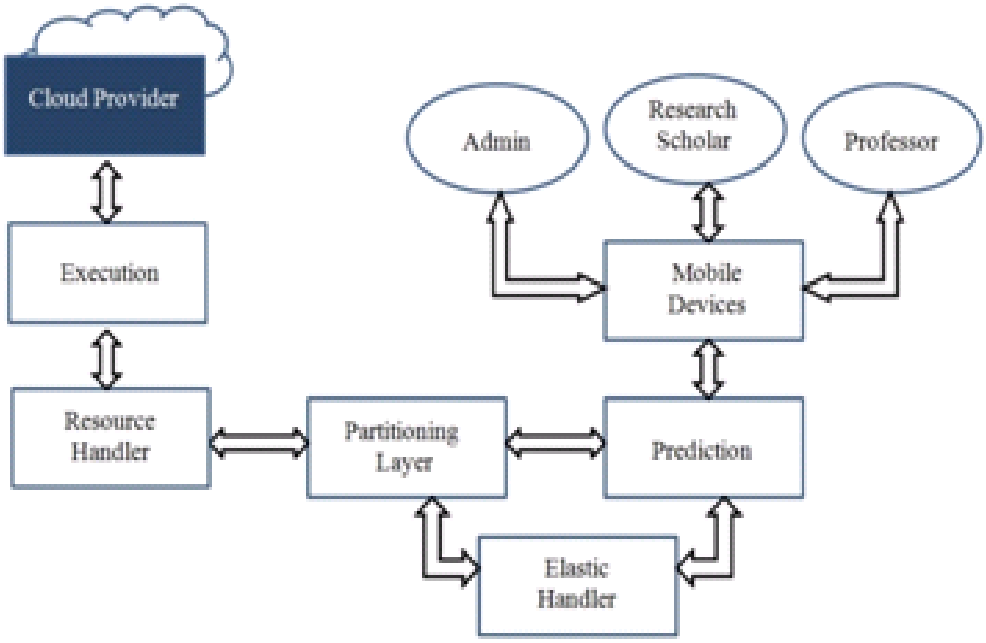

Fig. 2 Schematic diagram of partitioning elastic application

The work flow of partitioning the elastic application is represented in Fig. 2.

Stage 1: Admin, research scholar and professor act as an end user to access data in university cloud through their mobile devices.

Stage 2: Once a user decides to utilize the resource in the form of application, the process is initiated by using prediction technique in the cloud.

Stage 3: The partition layer helps to disintegrate the application. At this stage, the elastic handler acts as an intermediate between partitioning and prediction.

Stage 4: Resource handler takes care of resources on the cloud environment after partitioning.

Stage 5: After execution, any cloud vendor [16] can use this model to handle resources before it is delivered to end user.

3. RESULTS AND DISCUSSION

In this section, the experimental evaluation for the prediction based partitioning elastic model is discussed. This experimental model setup admits by varying customer access time, resource usage, size of partitioning module and scheduling policies. At first, the less number of users and system resources are tested and its result is shown in Fig.3. When there is increase in number of users and system resources the proposed model performs good when compared to the traditional partitioning technique which is shown in Fig. 4. The execution time of application based on the prediction approach is taken into consideration in both phases. With the help of the application prediction layer, the proposed model can dynamically modify the requirement of VMs and the resource allocation can effectively meet the requirement of system resources.

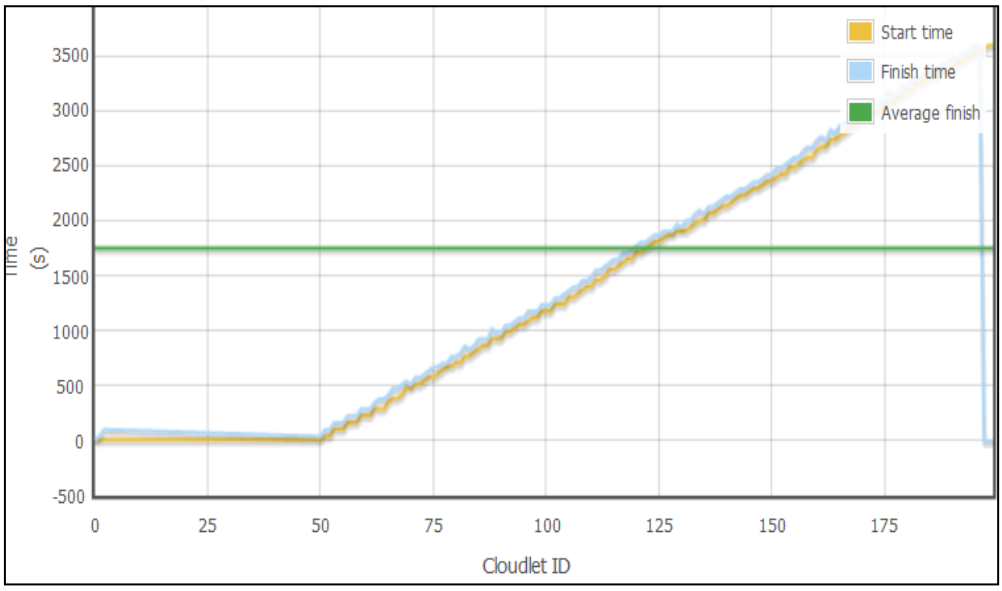

Fig. 3 Less number of users and resources

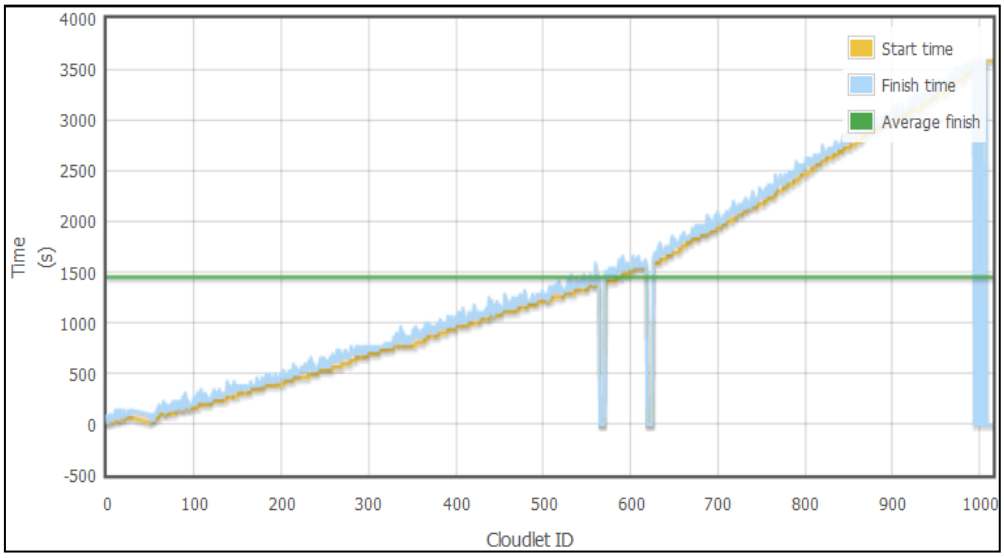

Fig. 4 High number of users and resources

4. CONCLUSION AND FUTURE ENHANCEMENT

Mobile Cloud computing paradigm is a better supported platform to enhance the interdisciplinary research activities in universities by collaboratively work with a pool of resources, on-demand, pay-per usage, elasticity. The related work examines the partitioning technique. To improve the elasticity, the partitioning technique is focused using cloud computing. To achieve this, a framework is proposed in this paper for partitioning the application in an elastic execution environment of cloud computing. Here, five layers support to improve its efficiency at the time of data transfer. The prominent feature of this framework is to improve the elasticity of application between mobile devices and cloud servers. In addition, the Univ-cloud using partitioning reduces the management cost and time when it is implemented for education purpose. In future, this work will be extended with offloading technique and resource scheduling management to improve the cost model and elasticity in mobile cloud computing.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing", Technical Report UCB/EECS, 2009.

[2] Le Guan, Xu Ke, Meina Song, Junde Song, "A Survey of Research on Mobile Cloud Computing", 10th IEEE/ACIS International Conference on Computer and Information Science, 2011.

[3] L. Yang, J. Cao, S. Tang, T. Li, and A. Chan, "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing", In Proc. of IEEE Fifth International Conference on Cloud Computing, 2012.

[4] K. Kumar, and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy", In IEEE Computer, 43(4), Apr. 2010, 51-56.

[5] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: a partition scheme", In Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, pages 1-6. IEEE Press, 2001.

[6] Byung-Gon Chun, SunghwanIhm, PetrosManiatis, MayurNaik, Ashwin Patti, "CloneCloud: Elastic Execution between Mobile Device and Cloud", In Proc. EuroSys'10.

[7] Chetan S., Gautam Kumar, K. Dinesh, Mathew K. and Abhimanyu M.A. "Cloud Computing for Mobile World", 2010.

[8] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing", IEEE Pervasive Computing, 8(4), 2009, 14–23.

[9] Zhiyuan Li, Cheng Wang, Rong Xu, "Task Allocation for Distributed Multimedia Processing on Wirelessly Networked Handheld Devices", In IPDPS'02.

[10] Ioana Giurgiu, Oriana Riva, DejanJuric, Ivan Krivulev, and Gustavo Alonso, "Calling the cloud: enabling mobile phones as interfaces to cloud applications", in Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware 2009, Springer-Verlag New York, Inc.: Urbanna, Illinois. 1-20.

[11] J. Rellermeyer, O. Riva, and G. Alonso, "AlfredO: An Architecture for Flexible Interaction with Electronic Devices", in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2008), ser. Lecture Notes in Computer Science, 5436. Leuven, Belgium: Springer, 2008, 22–41

[12] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. Tuulos, "Misco: a mapreduce framework for mobile systems", in Proc. 3rd International Conference on PErvasive Technologies Related to Assistive Environments. ACM, 2010.

[13] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, Ashwin Patti, "CloneCloud: Elastic Execution between Mobile Device and Cloud", In Proc. EuroSys'10.

[14] A. Messer, I. Greenberg, P. Bernadat, D. Milojicic, D. Chen, T. Giuli, and X. Gu, "Towards a distributed platform for resource-constrained devices", in Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, 2002, 43–51.

[15] Byung-Gon Chun and PetrosManiatis, "Dynamically partitioning applications between weak devices and clouds", in Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. 2010, ACM: San Francisco, California. 1-5.

[16] Gonzalo Huerta-Canepa, Dongman Lee, "A Virtual Cloud Computing Provider for Mobile Devices", In Proc. ACM MCS'10, pages 3756–3761. ACM Press, 2010.

[17] Eduardo Cuervoy, ArunaBalasubramanianz, Dae-ki Cho, "MAUI: Making Smartphones Last Longer with Code Offload", In Proc. MobiSys'10. ACM press, 2010.