Volume 4, Issue 10, October 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Shared Cache in Mobile Cloud Computing

[1]Soha Kasemi[*], [2]Dr. Kamran Zamanifar, [3]Dr. Naser Nematbakhsh
Department of Computer Engineering, Najafabad Branch , Islamic Azad University,
Najafabad , Iran

Abstract— Mobile Cloud Computing is a new technology that has been created along with the development of smart phones. However, compared to PCs, mobile devices are facing difficulties because of limitations in their hardware structure and servicing. For example, they have problems such as low bandwidth, low processing power, lack of memory storage and lower battery life. Therefore, we propose an architecture containing cloudlets put on the path between the user and the cloud server. We use cloudlets as a shared cache memory. To manage the content of cloudlets, then, we identify the quality parameters influencing the response time to users and based on it we design an algorithm that choose those items necessary for cloudlet. We compare the proposed algorithm with previous methods. Finally, we obtain an improvement of the response time by about 28% and 64% compared with a LFU algorithm and a case without cache memory, respectively. Therefore, less than before, we observe the dropping of data packets due to long latency.

Keywords— Mobile cloud computing , Cloudlet, Shared cache, Bandwidth

I. INTRODUCTION

Mobile cloud computing has been receiving much attention recently because it allows that storage and processing to be done outside of the mobile device. The growing popularity of this technology is due to the increasing number of smart phones that act like small PCs. The limitations of mobile phones such as low memory and low battery lifetime could be addressed by offloading of processing and storage in cloud memory [1]. Offloading operations could take place at a data center or even at a computer in the vicinity of mobile devices. Cloud computing is a framework for sharing resources, data, and softwares between different mobile devices. Resources are ready on clouds and can be shared upon request between mobile devices. In cloud computing environment, users can use the cloud to make backup copies of their mobile device. Although mobile devices including mobile phones and tablets have been much improved in terms of processing speed and internal memory capacity, they still have some shortcomings which decrease the performance of these devices compared to PCs. In addition to having a lower processing speed, memory capacity and battery life, the major challenge for mobile devices in cloud computing is data transfer bottleneck. To connect to the Internet, these devices adopt a way that may increase the response time of users due to low bandwidth and high traffic. On the other hand, battery is the main power source of these devices, but battery technology has not progressed to the extent that it can meet energy requirements fully. If cloud technology for mobile devices grows to the extent that many users are attracted to it, then the scalability in mobile cloud computing infrastructure becomes an important issue. In the absence of scalability solutions, this increasing growth creates a heavy burden on the network and produces an unacceptable response time [4].

II. LOW BANDWIDTH AND LOW-SPEED ACCESS OF USERS IN MOBILE CLOUD COMPUTING

Bandwidth is one of the big concerns in cloud computing because the sources of radio waves for wireless network is rare compared to traditional wired networks. For this reason, the available bandwidth is low and users have limited access speed to the information requested. In addition, mobile users may not be able to connect to the clouds due to traffic congestion, loss of network connection and not having signal. The cloudlet concept has been proposed by Satyanarayanan [3]. A cloudlet is a trusted, resource-rich computer or cluster of computers that's well-connected to the Internet and is available for use by nearby mobile devices. Therefore, when mobile devices don't tend to offload data to a cloud, they can find cloudlets and apply them. Offloading the data needed for processing and storage on a remote server will be completed by performing computations and sending the results to the mobile client. The different usable techniques for data offloading currently include client-server communication, virtualization and mobile agents. In these methods, it is possible for users to reach real-time requests, low latency, one hop solution and a fast access. If a cloudlet was not available nearby, mobile device refers to default state to deliver requirements into target cloud or at the worst case scenario only take advantage from its resources. In their article, Satyanarayanan et al. proposed an architecture using virtual machine technology so that they can create a sample of customized software service on a cloudlet nearby and then use that through Wireless LAN or WLAN. This technology can be used to help mobile users overcome the limitations of cloud computing such as latency and low-bandwidth through massive networks. However, before widely applying this system in practice, there are considerations that need to be addressed. For example, how can the network processing, storage and capacity be distributed for each cloudlet? How can policies for cloudlet service providers be administered to maximize the user's experience and at the same time to minimize the cost? Besides, it is an important issue to ensure the reliability and security for cloudlets in the implementation of this idea. The reason is that it is possible to create a fake cloudlet and to steal user data. Besides having many challenges in terms of security, this proposed scheme requires users to have access to powerful hardware [2]. In addition, the range of these small networks is limited and it is not always used.

Mobile cloud computing applications are widely used in many fields such as speech synthesis, natural language processing, image processing, information sharing, data searching and social networking. While many applications such as information sharing or social networking are not associated with fast processing, most focused applications such as image processing need fast processing and response. In another research, Joey et al. [5] have examined the shared cache memory. The shared storage approach tries to improve response time by reducing the composition time of virtual machine. If the users of cloud services have similar interests, shared storage significantly increases the response time. As an interesting application, a language translator is very useful for foreign travelers. Using mobile cloud computing, different words, sentences and paragraphs can be processed independently in the cloud. The most commonly used words and sentences will be available in the local cache memory which makes them more accessible in subsequent searches and improve the response speed. To improve the system response time, shared memory consists of several distributed memory. Distributed memories allow a system to deal with concurrent shared requests and shared contents. We can also reduce the response time by simultaneously retrieving data from different memory locations. The proposed scheme by Joy et al. [5] use a cache memory shared between several mobile phones and its storage system is based on latest user requests. In this way, a list of the latest users' requests is left in shared memory so that it can improve the speed of re-access to the information. The proposed scheme has been demonstrated to have good performance in the certain cases and can be used for users who have similar interests.

In this paper we provide a mechanism based on setting up a shared cloudlet which is not specific like Satyanarayanan's scheme and can meet the user's need to have a computer in parallel. But the best feature of this scheme is that it allows several users to use a cloudlet and also prevents users from interfering with data security. Thus, it is better for cloudlet to be put on the path of network platform under the authority of Internet service providers so that more users, in addition to preserve data security, are allowed to simultaneously use these features. Additionally, if the cloudlet service provider is the same Internet service provider, users will have less concern about the protection or loss of their data than they have in previously mentioned schemes. Having an agreement between cloud service provider and Internet service provider, cloudlet services can be shared between users with high security. In this study, we apply ideas from two above researches to present a cache memory which, through being located at an appropriate geographical position, is able to determine the contents of shared cache memory and to facilitate users' access to the information they need according to the algorithm designed. At first it is necessary to consider an appropriate position to locate the cache. The first important point about finding an appropriate position is to reduce geographical distance between cache memory and user compared to geographical distance between main cloud servers and user. To connect to the Internet, mobile devices must communicate with BTS masts just like when you're talking or you're using voice service. The data sent by users are transmitted to the communication center via BTS masts. Communication center is responsible for receiving wireless waves related to cell phones and transmitting them on the Internet. Data are then transmitted to the gateway via routers and eventually will reach their intended destination after passing through a route containing several switches and routers. The same route containing a large number of telecommunication equipments must be traversed to allow that response of received request is sent back and transmitted again to target user via BTS masts. If data packets don't be dropped over passing through the route, it will definitely cause a delay in the delivery of services to the user. Therefore, a mechanism is needed to reduce this problem somewhat.

If cloudlets are located in a position along the waves received via telecommunication masts on the network platform, transmitted data will not be required to divert their route and they will encounter with cloudlet in the route. Of course, this is only possible by means of an agreement done between cloud service providers and Internet service providers. We are going to turn a cloudlet into a cache memory and thereby to increase the response rate for users. Setting up a shared cloudlet service is expected to reduce significantly the steps to destination and thus to increase the response rate. The reason is that only a portion of the route by the user and the other by the cloudlet will be used and this is effective in access speed.

III. CACHE CONTENTS SELECTION ALGORITHM

To improve the response time to users, it is necessary to identify and evaluate the factors influencing the users' access speed to the information they need. Generally, several factors influencing this include: Free bandwidth on a communication path, the involvement rate of the cloud server CPU, the time to send the request as well as the amount of exchanged data. These factors will be evaluated using the monitoring software and the information obtained will be used for cache contents selection algorithm.

Initially, we establish a mathematical equation based on the factors influencing the response time. Based on this relationship, each item present in the cloud is assigned a numerical amount that represents the value of that item for storage in the cache memory space. G denotes this numerical value. The more larger is G, the more probability there is that corresponding data item to be stored in cache memory.

$$\text{Max } \{ G_{ij} = P_{ij} + Bw_j + C_j + S_i \}$$

In equation (1), G numerical value has been computed based on the parameters that affect the response time. The variables used include:

P_{ij} is the probability to request data item (i) in a few hours interval (j) on the day. In this study, a two-hour interval is considered. B_{wj} denotes the percentage of bandwidth occupied in a time interval (j) on the day. Cj represents the functionality percentage of cloud server CPU in a time interval (j) and Si is the volume percentage of the data item (i) relative to total exchanged volume in a time interval (j).

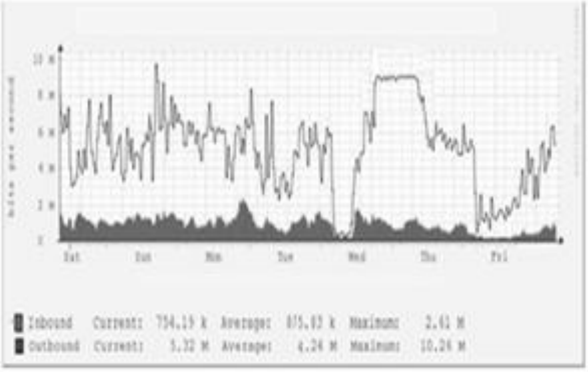

Figure 1 .Bandwidth in algorithm test

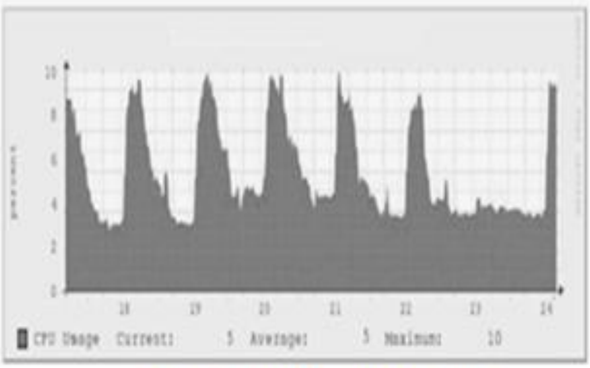

Figure 4 . CPU usage in this algorithm

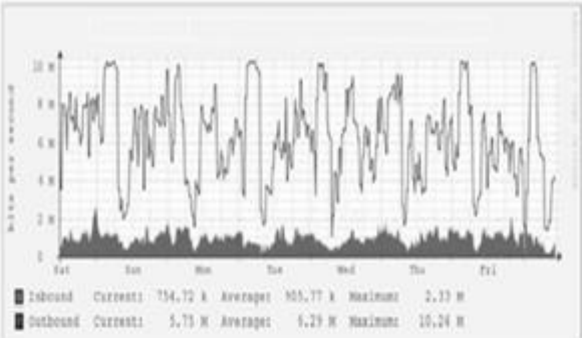

Figure 2 . Bandwidth in LFU algorithm

Figure 5 . CPU usage in LFU algorithm

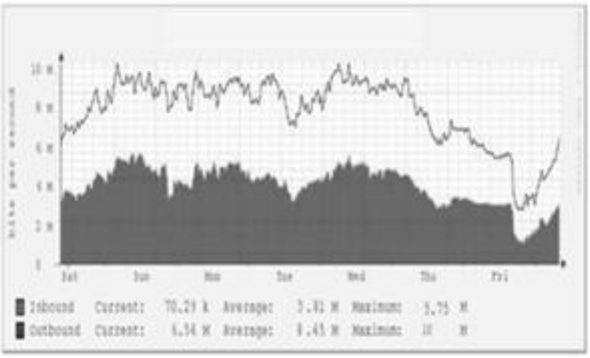

Figure 3 . Bandwidth without caching

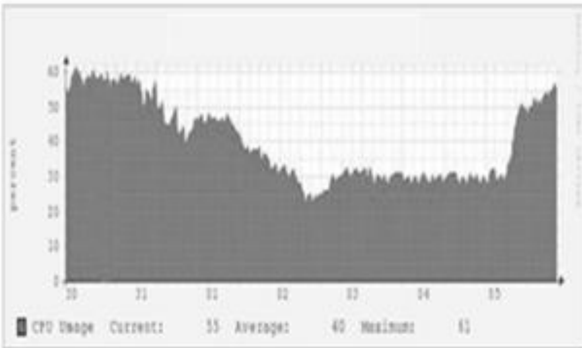

Figure 6 . CPU usage without cache

The algorithm in Figure 2 is written based on the above equation and Gs obtained are ranked in descending order. Each variable in the algorithm is assigned an impact factor that can vary depending on the implementation environment. The factors represent the importance of each variable. In general case, all factors can be assumed to be 1.

Consequently, those data items, which are requested at high traffic hours of the day (at a time when bandwidth usage and involvement rate of cloud server CPU are high) and occupy a high percentage of exchanged volume in an intended time interval, have a higher priority for storage in the cache.

Since the algorithm used in the shared cache memory employs the LFU algorithm, the above algorithm is implemented and the results are compared with two cases, including a case that use shared cache with LFU algorithm and a case without shared cache.

As can be seen in Figure 3, the bandwidth and CPU performance in the proposed algorithm have been increased significantly compared with both other cases.

The results from proposed algorithm testing in Figure 3 indicate that bandwidth is improved by 33% and 50%compared with LFU algorithm and a case without shared cache, respectively. Further, the CPU performance is improved by 44% and 88% on average compared with LFU algorithm and a case without shared cache, respectively. As a result, as expected, the response time to users is improved by 28% compared with LFU algorithm and by 64% compared with a case without shared cache.

IV. CONCLUSIONS

This study uses cloudlets as a cache memory to improve the access speed of servers by mobile cloud computing users. To this end, the distance between the mobile user and cloud server is divided into two parts and cloudlets are put on the path to cloud server. Therefore, it helped us to avoid creating additional overhead resulting from calculating the present distance to cache memory and main server. The cache contents then were selected based on proposed algorithm so that those data items which, in addition to high popularity, were requested at high traffic hours of the day, i.e. at a time when bandwidth usage and involvement rate of cloud server CPU were high, and meanwhile were large in size compared to the other items, had a higher priority for storage in the cache. Finally, a comparison between the proposed algorithm and LFU algorithm as well as a case without cache memory shows that bandwidth and cloud servers CPU have been improved. This means faster access to information requested by users and as a result, improvement of response time to users.

REFERENCES

[1] Mehta, Monal, Ishitva Ajmera, and Rakesh Jondhale. "Mobile Cloud Computing." *International journal of Electronics* (2013).

[2] Guan, Le, et al. "A survey of research on mobile cloud computing." *Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science. IEEE Computer Society,* 2011.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” *IEEE Pervasive Computing, vol. 8, no. 4, pp. 14-23,* October 2009.

[4] E. Jung, Y. Wang, I. Prilepov, F. Maker, X. Liu, and V. Akella, “User-profile-driven collaborative bandwidth sharing on mobile phones,”*in Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS), no. 2,* 2010

[5] Joy, Preetha Theresa, and K. Poulose Jacob. "Cooperative Caching Framework for Mobile Cloud Computing." *arXiv preprint arXiv:1307.7563* (2013*).*

[6] B. G. Chun, S. Ihm, P. Maniatis,M. Naik, and A. Patti, “CloneCloud: elastic execution between mobile device and cloud,” *in Proceedings of the 6th conference on Computer systems (EuroSys), pp. 301-314,* April 2011.

[7] A. Klein, C. Mannweiler, J. Schneider, and D. Hans, “Access schemes for mobile cloud computing,” *in Proceedings of the 11th International conference on mobile data management (MDM), pp. 387,* June 2010.

[8] Bharath, V., and K. Priyadarsini. " Enaergy efficient data traffic management in mobile cloud computing". *Energy 2.4*(2014).

[9] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, “Pocket cloudlets,” *in Proceedings of the 16th international conference on Architectural support for programming languages and operating systems (ASPLOS), pp. 171-184,* March 2011.

[10] Fernando, Niroshinie, Seng W. Loke, and Wenny Rahayu. "Mobile cloud computing: A survey." *Future Generation Computer Systems* 29.1 (2013): 84-106.

[11] Barbera, Marco V., et al. "To offload or not to offload? the bandwidth and energy costs of mobile cloud computing." *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013.

