



The Design of an Efficient Proxy-Based Framework for Mobile Cloud Computing

Zhijun Zhang^{1,2}, HyoTaek Lim^{1*}, and Hoon Jae Lee¹, *Member, KIICE*

¹Department of Ubiquitous IT, Graduate School, Dongseo University, Busan 617-716, Korea

²Weifang University of Science & Technology, Weifang 262700, China

Abstract

The limited battery power in the mobile environment, lack of sufficient wireless bandwidth, limited resources of mobile terminals, and frequent breakdowns of the wireless network have become major hurdles in the development of mobile cloud computing (MCC). In order to solve the abovementioned problems, This paper propose a proxy-based MCC framework by adding a proxy server between mobile devices and cloud services to optimize the access to cloud services by mobile devices on the network transmission, application support, and service mode levels. Finally, we verify the effectiveness of the developed framework through an experimental analysis. This framework can ensure that mobile users have efficient access to cloud services.

Index Terms: Active service, Cloud computing, Mobile cloud computing, Proxy server

I. INTRODUCTION

In recent years, with the rapid development of mobile communication technology as well as the increasing popularity of mobile devices, such as smartphones, the mobile network infrastructure, has continued to improve. Cloud computing services are no longer limited to the personal computer (PC) but extend to mobile devices. Based on smart terminals for cloud services to mobile devices, such as Apple's 'iCloud' service, Microsoft's 'live mesh' services, and Motorola's 'motoblur' service, a new computing mode, called mobile cloud computing (MCC), has been developed for mobile devices [1]. Juniper Research Company conducted a study from 2009 to 2014 and has reported that by 2015, the annual growth rate of mobile phone software based on cloud computing will reach 88% and the MCC market revenue will reach \$9.5 billion. Hence,

the MCC application marketplace is believed to have huge growth potential.

However, mobile devices have their own limitations with respect to the central processing unit (CPU) processing power, memory, and battery capacity; these constraints include limited resources, mobile wireless communication bandwidth constraints, and considerably expensive components. Further, most of the existing cloud services do not take into account the features of the mobile client, such as large quantities of data using the simple object access protocol (SOAP) protocol and resolution of the complex extensible markup language (XML) data format, which often leads to relatively large wireless network bandwidth resource consumption by MCC applications and expensive mobile equipment. Therefore, for the extension of the MCC application market, efficient and convenient access to cloud services by mobile devices is an urgent problem to be solved.

Received 26 March 2014, Revised 15 April 2014, Accepted 30 June 2014

*Corresponding Author HyoTaek Lim (E-mail: htlim@dongseo.ac.kr, Tel: +82-51-320-1718)

Division of Computer and Information Engineering, Dongseo University, 47, Jurye-ro, Sasang-gu, Busan 617-716, Korea.

Open Access <http://dx.doi.org/10.6109/jicce.2015.13.1.015>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

In this paper, we present a model of MCC based on intermediate proxy, in which a mobile terminal is connected to the cloud services architecture through an intermediate broker layer in order to realize a mobile environment and cloud service adaptation.

II. MOBILE CLOUD COMPUTING

There are several definitions of MCC available. In 2010, Google CEO Eric Schmidt described MCC in an interview by stating that on the basis of the development of cloud computing services, mobile phones will become increasingly complicated and evolve to a portable super computer [2]. We define MCC as follows: MCC is an emergent mobile cloud paradigm that leverages mobile computing, networking, and cloud computing to study mobile service models and develop mobile cloud infrastructures, platforms, and service applications for mobile clients. Its primary objective is to deliver a location-aware mobile service with mobility to users on the basis of scalable mobile cloud resources in networks, computers, storage devices, and mobile devices. Its goal is to deliver such a service with secure mobile cloud resources, service applications, and data using energy-efficient mobile cloud resources in a ‘pay-as-you-use model.’ Fig. 1 shows the conceptual architecture that reflects the definitions of MCC.

From the viewpoint of cloud computing, MCC is an organic combination of cloud computing and mobile computing. If cloud computing can be extended to various aspects of real applications, it will let people overcome time and space constraints; therefore, people will have the opportunity to use a variety of devices (smartphones, PDAs, tablets, etc.) anytime, anywhere, but not just confined to traditional fixed equipment such as PCs to gain quick and easy access to cloud computing and use it to provide various services, which fully embody the advantages of cloud computing.

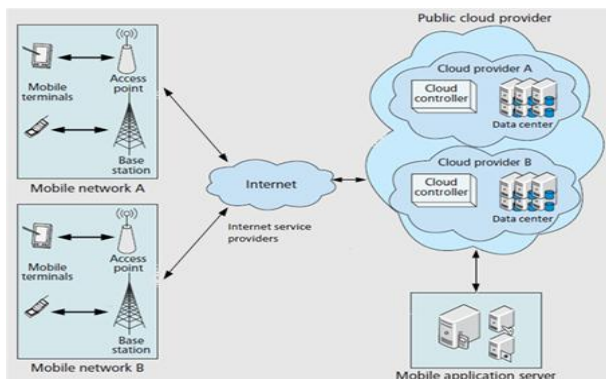


Fig. 1. Architecture of mobile cloud computing.

MCC is an extension of the expansion of mobile computing and cloud computing. In other words, MCC transfers the original intensive computing and large amount of information processing and storage from the mobile computing environment and processes it in a “cloud,” thereby changing the mobile application development, operation, deployment and usage patterns, and considerably reducing the consumption of the processing capacity of mobile devices and the resource requirements of mobile applications.

Mobile computing is the major driving force of MCC. Worldwide mobile application store downloads were forecasted to reach 17.7 billion downloads in 2011. This was a 117% increase from an estimated 8.2 billion downloads in 2010. According to Gartner Group, by the end of 2014, more than 185 billion applications will have been downloaded from mobile app stores [3].

With the rapid development of wireless network technology, represented by the smart phone of various mobile devices large number to access networks. According to one of the biggest risk investment institutions (KPCB, USA) in 2011, the global shipment of smart phone increased to 413 million, which was more than the total shipment of desktop and notebook computers [4].

MCC applications are applications that run on mobile devices in the frontend and on a cloud computing service in the backend. Their outstanding characteristic is that cloud computing in the backend is responsible for the main persistent storage calculations, such as tasks, while mobile devices are responsible for some simple calculations, such as show application interface. According to the application environment, MCC applications can be divided into browser-based Web applications and native applications (this article mainly refers to native applications). Currently, a vast majority of application developers can only take advantage of cloud service providers to provide a service interface for service access and return the corresponding results on a mobile device for processing and display. Since most of the existing service interfaces are not designed for mobile devices, the use of MCC applications often leads to larger mobile equipment resource consumption and wireless network bandwidth usage.

III. CHALLENGES AND SOLUTIONS

Mobile computing storage, batteries, and resource consumption. The processing power and memory of mobile devices were 2 to 8 times lower than those of a normal PC. This creates a bottleneck when mobile devices try to handle a complex process, such as XML data format analysis [5]. Therefore, to solve this issue, a mobile device can access cloud services to transfer a part of computing, storage, and processing operations to a powerful proxy server. Doing so

will decrease the consumption of mobile device storage, battery, and resource and increase the response speed of the service.

Wireless network transmission bandwidth restrictions. At present, XML is still the mainstream format used by cloud services for data transmission [6]. However, a large amount of data transmission does not adapt to the limited bandwidth of wireless transmission in the network environment. By using a proxy server, MCC application developers can create a bridge for data transmission format conversion, and consider using smaller volume data transmission formats (such as Java-Script Object Notation [JSON] and object serialization) for transmission, instead of XML, thus saving the cost of communication for mobile users [7].

Cloud computing service access restricted by resource limitations of a mobile device. Usually, mobile devices do not possess the ability to fit multiple cloud services in a mashup. Further, a mashup request of mobile devices needs to be established in every cloud service, which causes wastage of the resources of both mobile devices and wireless networks. MCC application developers should use proxy servers to establish a mashup of multiple cloud services and then, provide services to mobile devices; by doing so, they will ensure that mobile devices only have to set up a single connection request to multiple cloud services to complete the whole process.

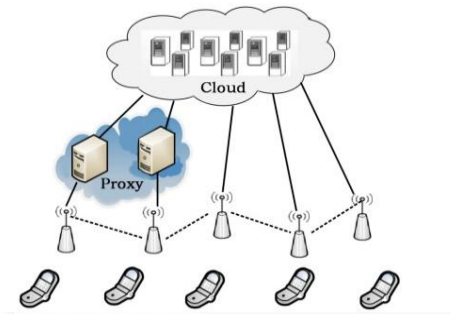


Fig. 2. Mobile cloud computing (MCC) framework based on proxy servers.

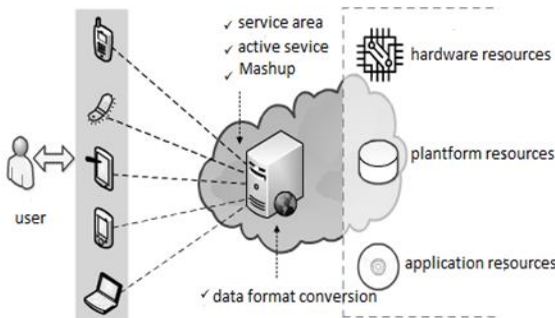


Fig. 3. Proxy layer function diagram.

On the other hand, as mobile computing has an inconsistent environment, mobile device users need to frequently change the settings related to cloud services according to the context information; this can at times lead to a service lag. Hence, a proxy server is used to solve this problem. After determining the situation of the mobile devices and collecting the relevant information, the proxy server will determine what services should be sent to different mobile devices according to the data collected. For example, the proxy server can suggest different types of applications to be downloaded depending on the scenario, thereby enabling the mobile devices to exhibit a better performance.

IV. MCC MODEL

In order to solve the problem stated earlier, in this paper, we propose an intermediate layer to join the middle layer of the MCC model, through the middle layer of the mobile devices and cloud services, to the proxy server in order to enable the access of mobile devices to cloud services, as shown in Fig. 2. Further, in the case of network transmission, there are three levels in the application support and service mode to optimize the access to cloud services. Therefore, the response of MCC application developers is faster, there is increased network traffic, and the application development framework is easier to use.

The intermediate broker layer needs to implement the following main functions: data format conversion, data transmission format from custom data transmission, remote service encapsulation, mashup and active services, etc., as shown in Fig. 3.

1) Data format conversion: Most of the existing cloud services are not designed for mobile devices. The intermediate broker layer converts the XML data to JSON data, which is relatively compressed and simple to parse [8].

2) Data transmission format from custom: Mobile users are usually interested only in a part of the response results of cloud services. However, they have to pay for the entire set of cloud service response results. Therefore, the MCC intermediate broker layer is used to support application developers, depending on the user's interest in the response results of cloud services for a custom operation in order to reduce the wireless network traffic.

3) Remote cloud service encapsulation: By encapsulating the underlying access detail of the distal cloud services, we can reduce the threshold of the MCC application development, allowing developers to put in more effort on the business logic of the MCC applications development.

4) Proxy services mashup: The middle layer of the mashup service and development of a uniform access interface for mobile devices can not only decrease the

intensive operations of mobile devices, storage, and battery, such as the consumption of resources, but also decrease the number of connection requests to multiple services, thereby decreasing the network traffic [9].

5) Active service: Through real-time monitoring the execution environment for mobile devices, and according to the changes in the environment, actively push services to mobile devices, so as to improve mobile users' experience.

From the above discussion, we can infer that an MCC middleware model based on agents is needed for developers to provide a simple, easy-to-use MCC application interface and a series of basic services, and to meet the functional requirements of cloud computing services for providing efficient and convenient access to mobile devices.

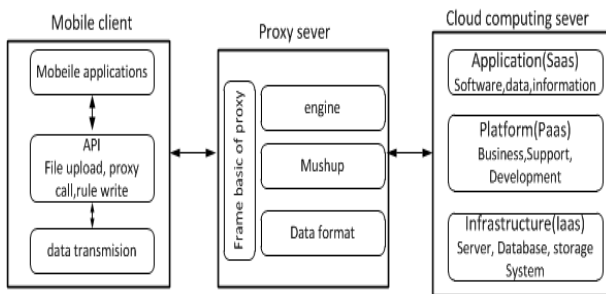


Fig. 4. Model architecture.

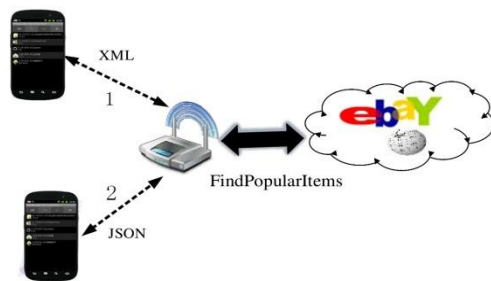


Fig. 5. XML and JavaScript Object Notation data tests.



Fig. 6. Middle server optimization test.

As shown in Fig. 4, a mobile application through which the client application development interface offers developers MCC application development tools, realizes the core functionality of the mobile cloud middleware based on proxy.

V. EXPERIMENTATION

A. XML and JSON Data Format

According to eBay (FindPopularItems API) of two different data format of the response as a result, we constructed two mobile applications access to it, as shown in Fig. 5. One describes a mobile device using the 1-Mbps bandwidth of the Wi-Fi access of the eBay FindPopularItems service and presents the service response results for the XML data format; the other describes mobile and mobile devices through the 1-Mbps bandwidth of the Wi-Fi access service response results of the eBay FindPopularItems service in the JSON data format.

Through the test, we can get the XML and JSON each service request response time, response the result processing time, request and response flow, the energy consumption of mobile devices as shown in Table 1.

We can infer from the table data that in mobile devices, with respect to the contents of the same cloud service access, the use of the JSON data transmission format can decrease the request and response flow by 30%, provide 17% faster service response time, improve the and response result by providing 47% faster processing time than that of the XML format. Mobile devices can decrease energy consumption by 25%. Compared with the XML data format, the JSON data format in the mobile transmissions has more advantages in the cloud computing environment.

B. Proxy Server

The use of proxy server middleware to access the XML format of cloud services resulted in optimized performance. To add the proxy server, we constructed two mobile applications access to Amazon (ItemSearch API) as shown in Fig. 6.

In Fig. 6, '1' describes a mobile device that has Wi-Fi access of 1-Mbps bandwidth to the Amazon ItemSearch service; the service response result is stored in the original XML data format. '2' describes mobile devices that have Wi-Fi access of 1-Mbps bandwidth to the Amazon ItemSearch service; the service response after the optimization processing by the proxy server are stored in the JSON data format, as shown in Table 2.

From the above analysis, we can conclude that mobile devices via the use of proxy server middleware to optimize

Table 1. Results of XML and JSON data access

Request service response time					
XML	975	963	997	973	1002
JSON	831	794	805	806	823
Response results processing time					
XML	1598	1541	1623	1558	1642
JSON	857	833	861	834	845
Request and response flow					
XML	22461	22466	22462	22464	22464
JSON	15670	15670	15672	15674	15674
Energy consumption of mobile devices					
XML	2.60	2.58	2.60	2.59	2.63
JSON	1.91	1.87	1.90	1.91	1.91

XML: Extensible Markup Language, JSON: JavaScript Object Notation.

Table 2. Results of XML and JSON data optimization

Request service response time					
XML	1533	1521	1543	1530	1542
JSON	7391	7371	7400	7395	7331
Response results processing time					
XML	10912	10897	7401	7394	7387
JSON	271	283	245	276	271
Request and response flow					
XML	61820	61821	61820	61822	61823
JSON	8701	8700	8695	8700	8701
Energy consumption of mobile devices					
XML	17.73	17.72	17.67	17.83	17.71
JSON	16.43	16.77	16.42	16.37	16.49

XML: Extensible Markup Language, JSON: JavaScript Object Notation.

the XML data format of cloud services access from direct access experience 92% savings on request and response flow, faster response results with a processing time of 9.89 seconds, and 18% savings on energy consumption; the delay between service request creation and response times is 4.85 seconds.

Through the above experiment, we can see that the use of proxy service middleware considerably simplifies the development of MCC applications. Thus, efficient access to cloud services is realized by MCC.

VI. CONCLUSIONS

With the rapid development of smart mobile terminals and cloud computing, a substantially increased number of users are expected to enjoy cloud services in the future by using mobile devices, and therefore, how to provide a better user experience has become an increasingly important research topic. In this paper, we discussed in detail, the

status of MCC in today's development scenario, described the related problems and challenges, and proposed a corresponding solution. Here, we proposed an agent-based MCC model that can provide access to cloud services by means of an intermediate proxy transport protocol and the data format of bridge transformation at the wireless network bandwidth. Experimental results show that this model is very effective. MCC is a research hot spot, and in the future, we intend to improve the user experience of mobile devices by enabling a more efficient access to cloud services.

ACKNOWLEDGMENTS

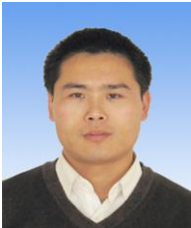
This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology. It was also supported by the BB21 project of Busan Metropolitan City.

REFERENCES

- [1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587-1611, 2013.
- [2] Z. Sanaei, S. Abolfazli, A. Gani, and M. Shiraz, "SAMI: service-based arbitrated multi-tier infrastructure for mobile cloud computing," in *Proceedings of the 1st IEEE International Conference on Communications in China Workshops (ICCC2012)*, Beijing, China, pp. 14-19, 2012.
- [3] 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 (EuroSys2011)*, Salzburg, Austria, pp. 301-314, 2011.
- [4] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 270-284, 2011.
- [5] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84-106, 2013.
- [6] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proceedings of IEEE INFOCOM*, Orlando, FL, pp. 945-953, 2012.
- [7] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: a partition scheme," in *Proceedings of the 2001 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES2001)*, Atlanta, GA, pp. 238-246, 2001.

[8] D. Kovachev and R. Klamma, "Beyond the client-server architectures: a survey of mobile cloud techniques," in *Proceedings of the 1st IEEE International Conference on Communications in China Workshops (ICCC2012)*, Beijing, China, pp. 20-25, 2012.

[9] H. Qi and A. Gani, "Research on mobile cloud computing: review, trend and perspectives," in *Proceedings of the 2nd International Conference on Digital Information and Communication Technology and It's Applications (DICTAP2012)*, Bangkok, Thailand, pp. 195-202, 2012.



Zhijun Zhang

received his B.S. in Computer Science and Technology from Xi'an Aeronautical University, Xi'an, China, in 1999, and his master's degree in Computer Application Technology from Ocean University of China, Qingdao, China, in 2012. He is currently pursuing his Ph.D. at Dongseo University, Busan, Korea. His research interests include network computing and network mobile cloud computing.



HyoTaek Lim

received his B.S. in Computer Science from Hongik University, his M.S. in Computer Science from POSTECH, and his Ph.D. in Computer Science from Yonsei University in 1988, 1992, and 1997, respectively. From 1988 to 1994, he worked for the Electronics and Telecommunications Research Institute as a researcher. Since 1994, he has been with Dongseo University, Korea, where he is currently a professor in the Division of Computer and Information Engineering. His research interests include computer networks, protocol engineering, storage networking, and cloud computing.



Hoon Jae Lee

received his B.S., M.S., and Ph.D. in Electrical Engineering from Kyungpook National University in 1985, 1987, and 1998, respectively. He was engaged in research on cryptography and network security at Agency for Defense Development from 1987 to 1998. Since 2002, he has been working for Department of Computer Engineering of Dongseo University as an associate professor, and is currently, a full professor. His current research interests include security communication systems, side-channel attacks, USN, and RFID security. He is a member of a number of educational organizations and societies such as the Korea Institute of Information Security and Cryptology, IEEE Computer Society, and IEEE Information Theory Society.