Development Model and Environment for Dynamic Mobile Cloud Services

Kuo-chu Wu
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan, ROC
Email:wukauzun@gmail.com

Wei-y Liu
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan, ROC
Email: m9921033@gmail.com

Shiow-yang Wu
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan, ROC
Email: showyang@mail.ndhu.edu.tw

***Abstract*—Many research and forecasting groups have suggested that mobile cloud computing is the next big thing in IT industry. The technologies for the widespread offering of mobile cloud services, however, are still significantly lagged behind. We propose an elastic framework and the associated application development model for effective application development and highly dynamic service provisioning in mobile cloud environments. To overcome the challenges of limited resources, low wireless bandwidth and intermittent disconnection, our framework enable end service users to smoothly access mobile cloud services by employing all available resources on all possible platforms including cloud data centers, base station servers, client devices, and even peer devices. With our semi-automatic development model and environment, service developers only need to provide necessary components for device-side mobile service interface and cloud-side service content. The service development middleware supports automatic service compilation, multi-platform service generation and service deployment on different environments.**

***Index Terms*—-cloud, mobile cloud, service migration, development model**

I. INTRODUCTION

According to Canalys survey, in the year 2011, venders shipped 488 million smart phones which surpassed 415 million of PCs for the first time [1]. With such an unprecedented growth rate, there is little doubt that mobile devices will soon become the main information access device for users around the world. Even with constantly improving device hardware and mobile networking systems, there are certain deficiencies and limitations. Mobile devices will always be resource-poor, less secure, with unstable connectivity, and with less energy since they are powered by battery [2]. To overcome these obstacles, both industrial experts and academic scholars have proposed many possible solutions. A particular promising one is the use of cloud computing to enhance the functionalities of mobile devices which has laid the foundation for a novel computing model, called mobile cloud computing, allowing users anytime anywhere access to unlimited computing power, storage space and online services. From a recent report by IDC, cloud computing, mobile-centric applications and interface, as well as media tablets are listed among the Top 10 Strategic Technologies for 2012 [3]. The forecast by IDC also indicates that mobile, cloud, and social networking are becoming the main stream in world IT market. ARCchart even predicts that by 2015, mobile generated contents will take up to 9,400PB (petabytes) of the cloud storage space [4]. The common theme of these forecasts and predictions all point out the importance and high market potential of mobile cloud computing.

However, the integration of the two technologies is far from straightforward. The CPU, memory, screen and other resources on mobile devices are relatively poor in comparison with their desktop counterparts. They are unlikely to be able to host all variety of cloud services smoothly. Another intrinsic characteristic of mobile networking is the intermittent disconnection and the need for disconnected operations. This is clearly against the always-on requirement of cloud services. It is therefore necessary to develop highly flexible mobile cloud service architecture and service management technologies to dynamically partition and seamless integrate the services among cloud, base stations and mobile devices. Furthermore, the development of mobile cloud services in such environments is highly challenging. We propose an elastic framework and the associated application development model for effective application development and highly dynamic service provisioning in mobile cloud environments. Our framework supports the idea that the users must be able to smoothly access mobile cloud services by employing all available resources on all possible platforms including cloud data centers, base station servers, client devices, and even peer devices. This is achieved by a highly dynamic service provisioning architecture which can switch the actual service platform at run time based on current resource availability and connectivity. We also provide semi-automatic application development model and environment such that it is relatively easy for service developers to write programs for our framework. More specifically, then only need to provide necessary components for device-side mobile service interface and cloud-side service content. The development middleware supports automatic service compilation, multi-platform service generation and service deployment on different environments. We have actually implemented the necessary modules and middleware for supporting the development and execution of application services in our framework. Preliminary evaluation results demonstrate that the proposed architecture and models are feasible and effectively realizable with very low overhead.

The rest of the paper is organized as follows. Section II provides a survey of related issues and research work. Section III presents the proposed dynamic service provisioning framework and development environment. Section IV provides detail account of the semi-automatic service development model. In Section V, we outline the prototype implementation of our ideas and present the results of performance evaluation on different dimensions of the proposed techniques. Section VI concludes the paper.

II. Related Work

The term "cloud computing" was first academically used by Professor Ramnath K. Chellappa to describe a computing paradigm bounded only by rationale, not by technologies. The so-called "Berkeley view of cloud computing" emphasized not only the cloud services but also the hardware and systems software that provide those services [5]. The NIST offered a comprehensive definition which is composed of five essential characteristics, three service models, and four deployment models [6].

Mobile devices are increasingly becoming an essential part of human life as the most effective and convenient communication tools not bounded by time and place. However, mobile computing is highly challenging with issues such as battery life, storage, bandwidth, mobility and security [7]. All these issues remain to be solved in mobile cloud environments. Jin and Kwok propose a solution to share limited bandwidth for cloud assisted P2P media streaming [8]. To support continuous operation when disconnected from the cloud, Huerta-Canepa and Lee propose to explore the connectivity of neighboring nodes to form a virtual cloud. Similarly, when infrastructure based access to the cloud is unavailable for some mobile users, the MoNet system and the WiFace application proposed by Zhang et al. employ multi-hop MANET to regain access with the help of co-located partners in an ad hoc manner [9].

The unique characteristics of mobile cloud services entail a smooth service provisioning architecture to integrate mobile applications and cloud services, especially when the users are accessing the services on the move. Takasugi et al. propose an overlay network technology to provide seamless-service environment for mobile users[10]. Rashid and El-Darieby introduce the Service Roaming Protocol (SRP) to manage service mobility [11]. Larosa et al. employ a mobile P2P architecture on heterogeneous wireless networks to provide reliable mobile cloud services [12]. The support of Quality of Service(QoS) is essential in mobile cloud environments. Peng et al. employ fuzzy cognitive map to develop an adaptive QoS-aware system to monitor the QoS state of each mobile cloud service terminal and the key parameters to guarantee the QoS [13].

The lack of good application development model and environment is another key that may hinder the wide spread offering of mobile cloud services. The OdinTools developed by Meads and Warren offer a model-driven middleware for mobile services [14]. Andreas et al. suggest access schemes to overcome data transfer compatibility problems in heterogeneous network for mobile cloud computing [15].

III. Elastic Frame work for Mobile Cloud Services

Most cloud services presume steady connection to the cloud providers with little or no consideration of bandwidth issues. Mobile cloud services, however, must consider the instability of wireless communications and the heterogeneity of the operational environments. It is also desirable for mobile cloud services to exploit all resources and equipments (such as the cloud data center, the business servers, base stations, mobile devices, etc.) currently available to the user. This calls for a highly dynamic service provisioning architecture that can automatically choose the best execution environment for a mobile user. We propose an elastic framework for mobile cloud services as illustrated in Figure 1. We argue that a service for mobile cloud must be designed in such a way that it can be smoothly partitioned, dynamically downloaded and continuously executed in various environments. When the connection quality is good and stable, the entire service can be executed in the cloud as normal. In case of unstable networks or intermittent disconnection, it is desirable to download the entire service(or service component) to the closest platform with stable connection to the user. When no such platform exists, it should even be possible to directly execute the service on the mobile device. When the user moves to a new location, seamless service migration must be provided such that the service provisioning remains uninterrupted. Service/data prefetching and buffering mechanisms help service provides taking initiatives to assist mobile users in quickly getting into online service environment. The above mentioned dynamic and flexible service provisioning framework cannot be achieved without the full coordination from all parties including cloud servers, base station servers, local servers and mobile devices. To fully support disconnected operation, we even incorporate peer-to-peer cooperation mechanism between mobile devices (P2P client coordination), so that the service may be provided by nearby peer devices which have already downloaded the service.

In order to achieve the goal of high-performance mobile cloud services, the first and most important step is to establish a highly elastic mobile cloud service architecture and associated dynamic service management mechanism. We propose an elastic mobile cloud service architecture design as depicted in Figure 2.

If the wireless connection is stable, the user can obtain services directly from the cloud center. If the connection is unstable or when the providers want to improve service performance, the *local service coordinator* of the base station where the user resides can take proactive measures through the *dynamic service downloader & partition manager* to download the current service or component to the *local service cache*. When the user is about to get out of the current cell, the *pre-fetching manager* can proactively prepare the target service or component by downloading it to the next cell that the user is about to visit. If the user cannot obtain needed services from base station server, the *P2P server* the cloud or the current *coordination manager* will try to locate the services from neighboring base station servers. During the service operation, the users may pass through several cells or base station servers. The *elastic service migration manager* ensures seamless service mobility. *Service status monitoring & QoS manager* is

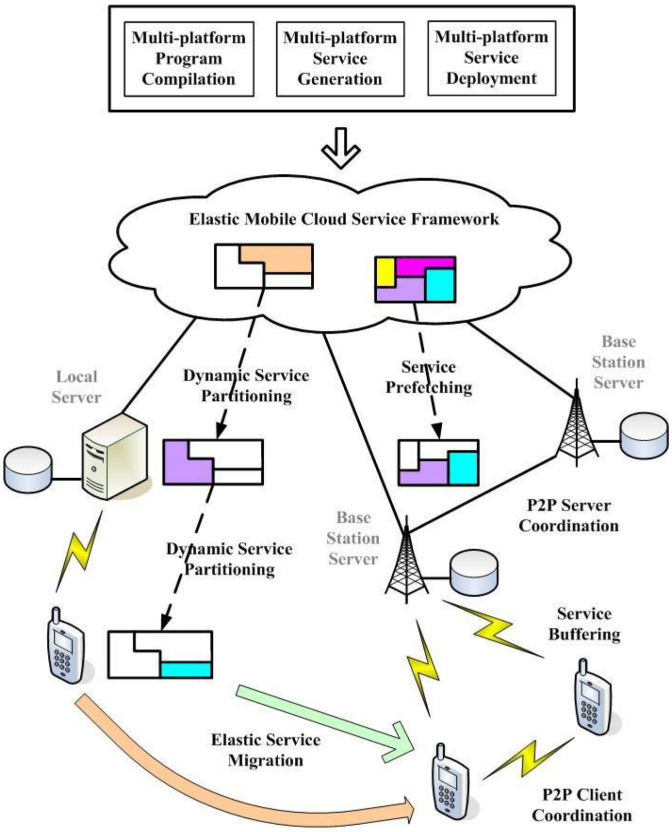

Fig. 1 Mobile cloud service framework

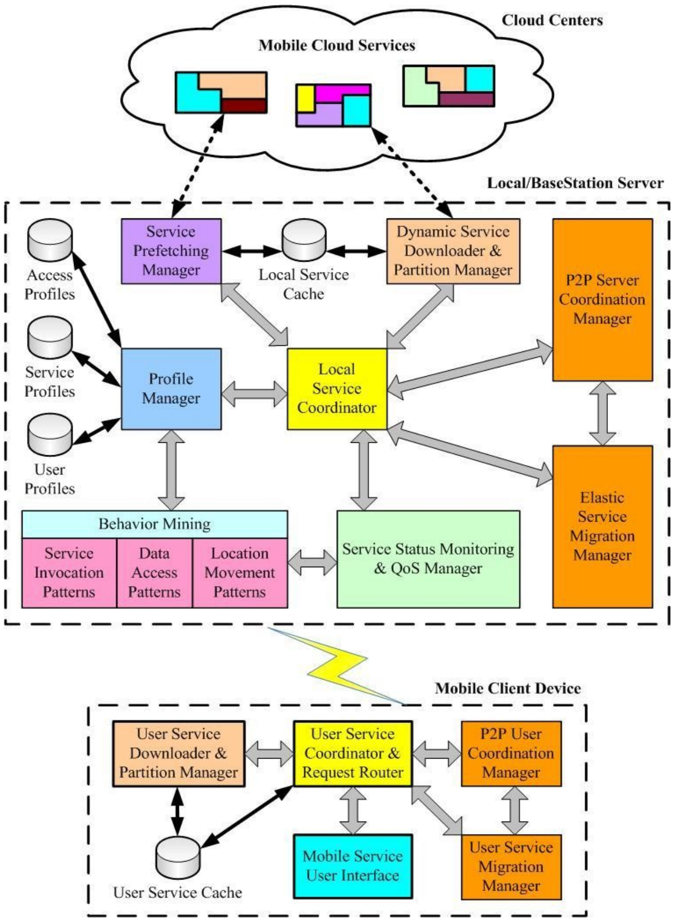

Fig. 2 Elastic mobile cloud service provisioning architecture

responsible for recording full service status and service quality changes, in order to provide local service coordinator some basis for decision making about service provisioning and server collaboration. A set of profiles are maintained by the *profile manager* and processed by the *behavior mining* module to establish user behavior patterns. These information help the coordinator in making prefetching, caching and coordination decisions.

The mobile client devices must have corresponding service management modules in order to allow the smooth operation of the whole architecture. The *user service downloader & partition manager* is responsible for coordinating with cloud or base station servers to download the necessary service or component to the *user service cache* to support disconnected operation. With our flexible architecture design, a service may be provided by the cloud center, base station server, peer device or service cache, the *user service coordinator & request router* is responsible for submitting service requirements and related parameters to the right service providing sites. When the service is from a peer device, the *P2P user coordination manager* is to coordinate with peer device on service provisioning. To maintain stability in the mobile service operations, the *user service migration manager* must work closely with the corresponding server-side module for seamless service mobility when the user is moving between cells. All management functions are carried out in the background. User interacts with the device through the *mobile service user interface*. Middle services are completely transparent to the user who can simply enjoy flexible mobile cloud services with high-quality experience.

IV. Development Model and Environment for Dynamic Mobile Cloud Services

As mentioned earlier, the development of mobile cloud applications to allow dynamic service provisioning is highly challenging. In addition to modularity, it is necessary for the modularized services to be able to execute in a variety of heterogeneous platforms. If the programmers are required to develop and port the same service on various platforms, it will be very time-consuming and error-prone. It is not advantageous for developing mobile application service market. We provide a multi-platform mobile cloud service development environment so that service developers only need to provide the necessary mobile service user interface and the cloud service function. Through our multi-platform program compilation, service generation and service deployment mechanisms, the system can automatically generate highly flexible mobile cloud services which can be smoothly executed in the architecture discussed in previous section.

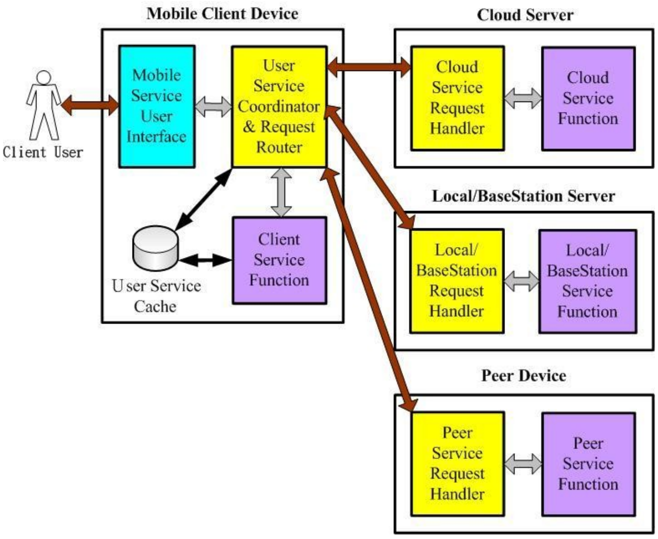

Fig. 3. Execution model for elastic mobile cloud services.

Based on the service architecture discussed in previous section, a simplified execution model for the mobile cloud services is depicted in Figure 3. The *mobile service user interface* handles the interactions between mobile service and end user. The interface should be installed and executed on the *mobile client device*. Users' instructions and data are packed and processed by the *user service coordinator & request router* which is responsible for determining and routing the messages to appropriate destination for execution. Each type of destination has a corresponding *request handler* to accept the client messages and sent to the corresponding *service function* module where the service is offered. The *client service function* represents the part of the service that has already been downloaded to the client device. In such case, the service request is routed directed to the client service function for local processing. This usually happens when the client is under disconnected operation.

Based on the execution model above, it may seem like the service developers need to provide all the components to enable dynamic service provisioning. To alleviate programmers from such a burden, we provide a development model and the associated environment that could automatically generate most of the components to make it as easy as developing a normal cloud service with mobile user interface. More specifically, the programmers need to design just the interface and main service function by following our development model. Then, by following our development process, all other components can be automatically generated without user intervention. Using Android system as an example, we explain the development process of the client interface as illustrated in Figure 4. The discuss below presumes the knowledge for a normal Android APK development.

First of all, the developers need to define the graphic interface for mobile users. For Android system, the interface is defined in the `layout`. User interface data are obtained through

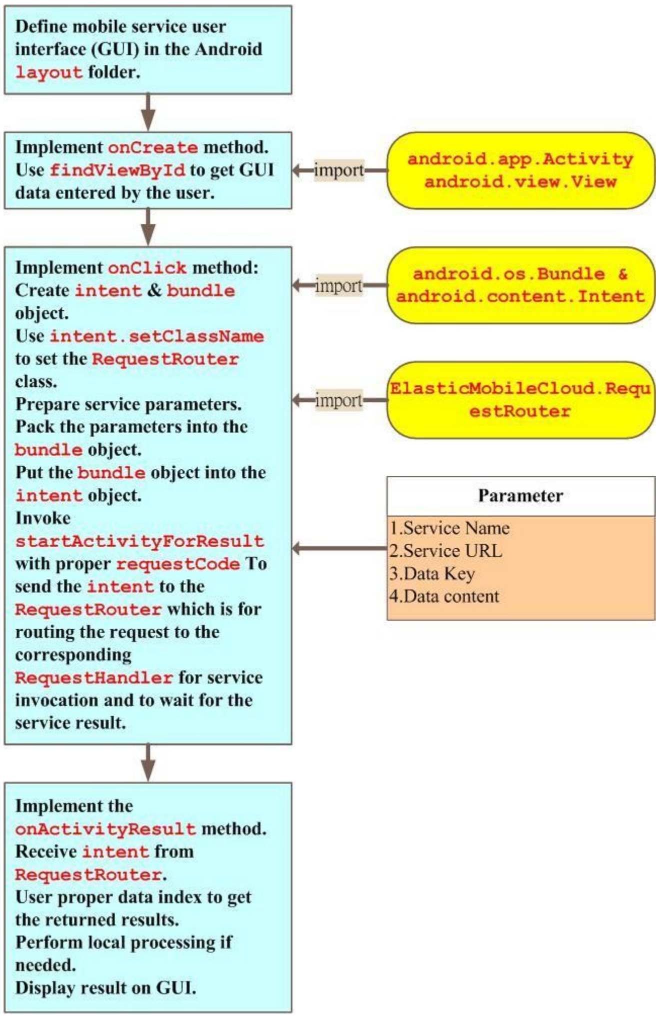

Fig. 4. Android user interface development model.

`findViewByld` in the `onCreate` method when end users click or input data in the client interface. The proposed *user service coordinator & request router* is implemented as an object from the `ElasticMobileCloudRequestRouter` class. The client program should define two objects, `intent` and `bundle`, which are saved in the `onclick` method. The `intent` should set the receipt port as the `requestRouter` class. The `bundle` should pack all the data sent from the interface. After packing the data, the `intent` should send out the data to the destinations, through `startActivityForResult`. The *user service coordinator & request router* will wait on `ActivityResult` until the results are sent back by a request handler. The client program then uses `ActivityResult` to receive the results, and the `intent` object to get the final results. The results are processed and displayed on the GUI in the same way as a normal Android APK. We emphasize that the developers only need to design mobile user interface by following the process above. The system and middleware we developed automatically complete the rest--service platform selection, service messages handling, service calls, and service result receiving. This greatly reduces the burden on the developers.

Another module that must be provided by the programmer is the main service functions which also need to follow the development model we suggested. Take Java as an example, the service development model is shown in Figure 5. The model is supported by a series of batch processing process. The developer use Java to develop the core service functions. Then with the *multi-platform service project creation batch processing* process, the service functions are transformed into projects that are suitable for execution on cloud server, base station server, peer device, and client device respectively. Each project is sent through the corresponding service compilation template to generate a proper compilation process. The *multi-platform service compilation batch processing* process is then used to generate all versions of the service ready for deployment. The compiled projects is then deployed to the cloud server and the other platforms.

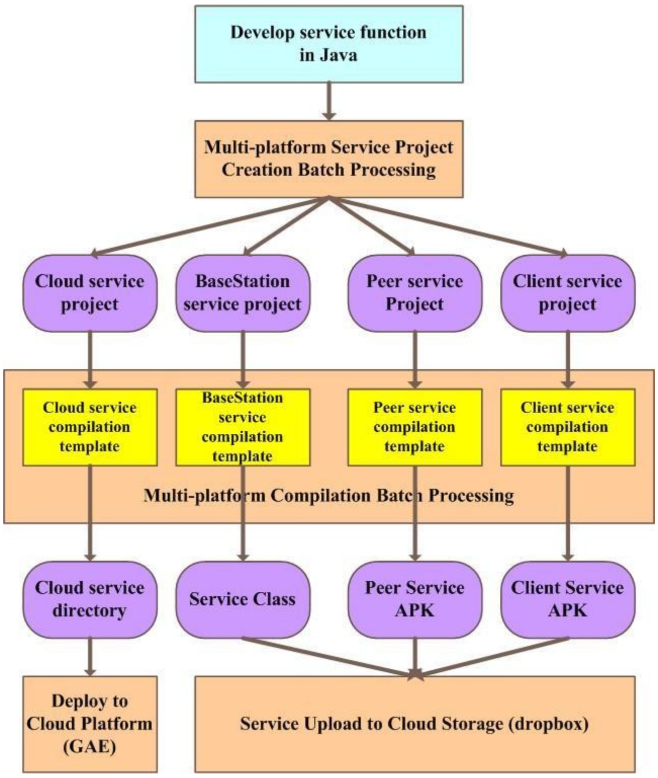

Fig. 5. Mobile cloud service development model

V. Performance Evaluation

To understand the efficiency of our service developing and provisioning mechanisms, we first develop a simple picture-downloading service by following our development model. We then test the performance of the automatically generated services on a variety of different platforms. The service is to download pictures of sizes from 500K to 5M. We then evaluate the performance of executing the service on the cloud server (Google App Engine), base station server, and peer device, as demonstrated in Figure 7. We use an Android mobile phone to download 20 times for each image, and calculate the average download speed. The result is shown in Figure 8. It can be observed that all generated services function properly and the processing time grows linearly with the image size on all platforms. It is also interesting to note that downloading from a

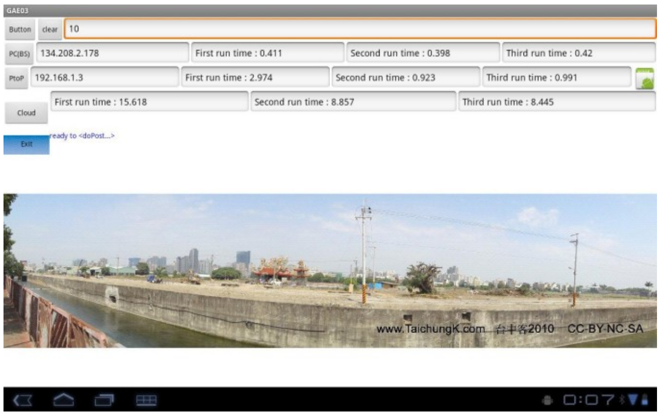

Fig. 7. Evaluation system user interface

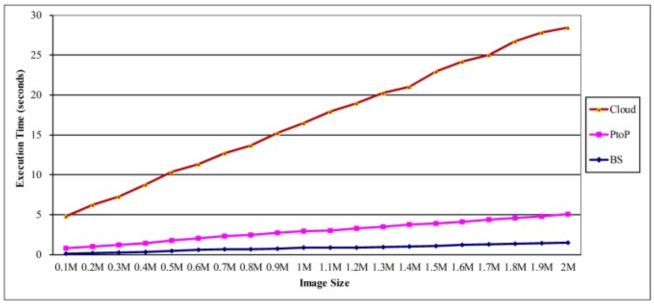

Fig. 8. Image download time on various platforms

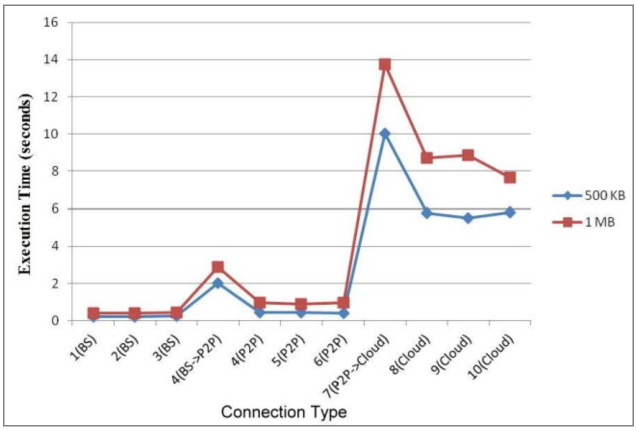

Fig. 9. Dynamic service provisioning on small images

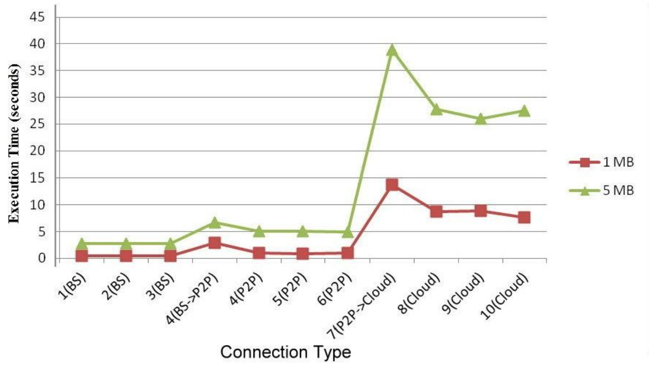

Fig. 10. Dynamic service provisioning on large images

nearby base station server or peer device is faster than that from the cloud. This also brings us addition benefits on bring services closer to the client.

To evaluate the responsiveness of our system on dynamic service provisioning and smooth switching between different platforms, we conduct an experiment to keep downloading pictures while the environment changes. Initially, the service downloads pictures from the fastest base station server(BS). Then we simulate the disconnection from the base station which triggers the switching to the peer device (with the service installed) as the next best choice(P2P). When the peer is again unavailable, we switch to the cloud for service. Figure 9 demonstrates the results using 500KB and 1M sized pictures while Figure 10 demonstrates similar results with 1MB and 5M sized pictures. It can be observed that the service maintains stable performance on each platform except at the time of transition(step 4 and 7). At step 4, when the base station service is no longer available, the system switches to P2P. At step7, when P2P service is again unavailable, the system transfers to the cloud. In both cases, higher delay can be observed due to timeout processing and service transition. Nevertheless, the service delay quickly goes down to normal and stable state at the next step.

VI. Conclusions and Future Work

In this paper, we have developed and demonstrated the feasibility and effectiveness of an elastic framework for highly dynamic service provisioning in mobile cloud environments. For developing services in the framework, we offer a development model and environment to assist service developers who only need to provide necessary components for device-side mobile user interface and cloud-side service function. The system automatically generate proper services for the deployment and invocation on different platforms. The run-time system also support smooth transition of a service from one platform to another when the former is unavailable. Evaluation results demonstrate that the generated services maintain stable and effective performance under stable connection. In the case when a service transition is required, the system can quickly locate the next available service platform with limited transition delay.

We are currently working on the mechanisms for automatic service partitioning, service prefetching, service buffering and service migration. With proper and effective solutions to each components, the proposed framework should offer a solid and responsive basis for mobile cloud computing.

Acknowledgment

This paper is partially supported by National Science Council, and project number is NSC-101-2221-E-259-014.

References.

[1] Canalys, “Smart phones overtake client pcs in 2011,” Canalys, Tech. Rep. Press release 2012/021, Feb 2012.

[2] R. A. Bajad, M. Srivastava, and A. Sinha, “Survey on mobile cloud computing,” *International Journal of Engineering Sciences & Emerging Technologies*, vol. 1, no. 2, pp. 8–19, Feb 2012.

[3] F. Gens, “Top 10 predictions - idc predictions 2012: Competing for 2020,” IDC, Tech. Rep. IDC #231720, Dec 2012.

[4] ARCchart, “The mobile cloud: Market analysis and forecasts,” ARCchart, Tech. Rep. SKU: CCEQ6428782, Jun 2011.

[5] 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,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[6] P. Mell and T. Grance, “The nist definition of cloud computing,” National Institute of Standards and Technology, NIST Special Publication 800-145, Sep 2011.

[7] M. Satyanarayanan, “Fundamental challenges in mobile computing,” in Proceedings of the 5th annual ACM symposium on Principles of Distributed Computing, May 1996, pp. 1–7.

[8] X. Jin and Y.-K. Kwok, “Cloud assisted p2p media streaming for band-width constrained mobile subscribers,” in Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems, ser.ICPADS ’10, 2010, pp. 800–805.

[9] L. Zhang, X. Ding, Z. Wan, M. Gu, and X.-Y. Li, “Wiface: a secure geosocial networking system using wifi-based multi-hop manet,” in Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ser. MCS ’10, 2010, pp. 3:1–3:8.

[10] K. Takasugi, M. Nakamura, S. Tanaka, and M. Kubota, “Seamless service platform for following a user’s movement in a dynamic network environment,” in Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, ser. PERCOM ’03, 2003, pp. 71–78.

[11] B. Rashid and M. El-Darieby, “Towards seamless service mobility for mobile devices communicating within wireless grids,” in Canadian Conference on Electrical and Computer Engineering, CCECE 2007, Apr 2007, pp. 491–495.

[12] Y. Larosa, J.-L. Chen, D.-J. Deng, and H.-C. Chao, “Mobile cloud computing service based on heterogeneous wireless and mobile p2p net-works,” in Wireless Communications and Mobile Computing Conference (IWCMC), 7th International, 2011, pp. 661–665.

[13] P. Zhang and Z. Yan, “A qos aware system for mobile cloud computing,” in Cloud Computing and Intelligence Systems (CCIS), IEEE Interna-tional Conference on, 2011, pp. 518–522.

[14] A. Meads and I. Warren, “Odintools–model-driven development of intelligent mobile services,” in Services Computing (SCC), IEEE In-ternational Conference on, 2011, pp. 448–455.

[15] A. Klein, C. Mannweiler, J. Schneider, and H. D. Schotten, “Access schemes for mobile cloud computing,” in Mobile Data Management (MDM), 11th International Conference on, 2010, pp. 387–392.